november/december 2018 | Nr. 650 | www.elektormagazine.nl | € 14,95

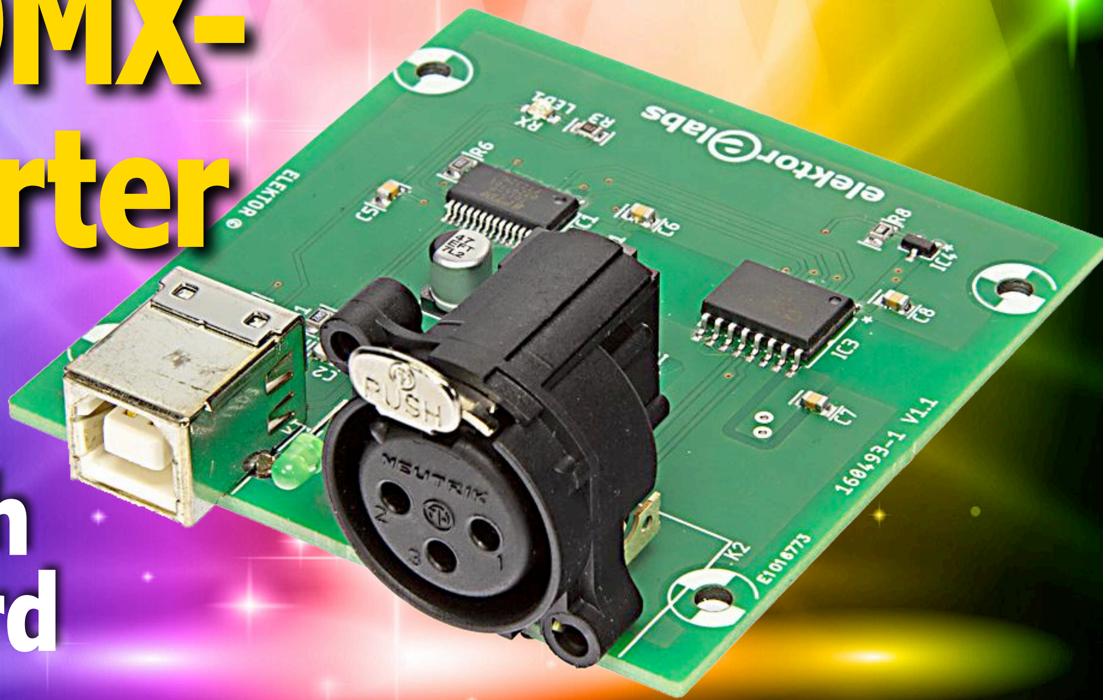
elektor

elektorlabs magazine

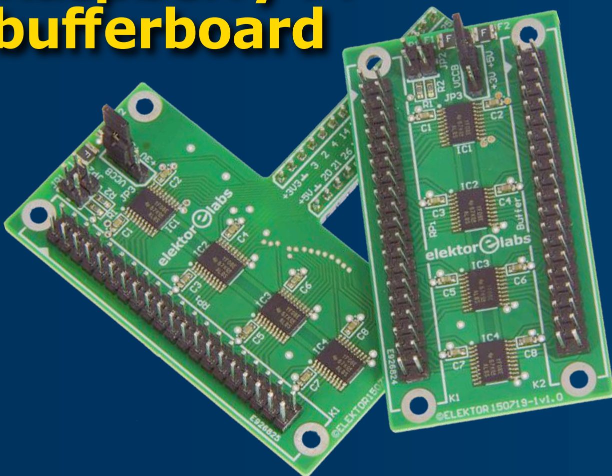
DESIGN ⊳ BUILD ⊳ SHARE **ELECTRONICS**

Compacte USB/DMX-Converter

volledig elektrisch geïsoleerd

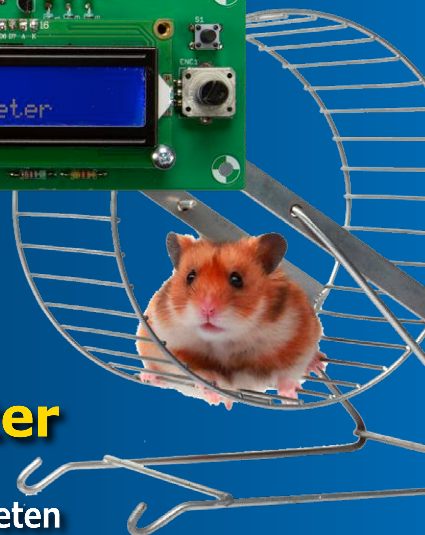


Raspberry Pi bufferboard



nooit meer opgeblazen I/O's





Hamster Run-O-Meter

km/u en afstand meten

Ultra-low-power ontwerp ⊳ SDR-shield als meetinstrument ⊳ Opamps in de praktijk ⊳ Kortsluiting-zoeker ⊳ ESP8266 in de modelbouw ⊳ Kleuren-OLED's ⊳ Magische lamp ⊳ DDS met Arduino ⊳ Actief audio-wisselfilter ⊳ AVO buizentesters ⊳ Hardware-ontwerp met (V)HDL ⊳ ...en meer!

BP 8 710966 982907 00650

Colofon

58e jaargang nr. 650,
november/december 2018
ISSN 0013-5895

Elektorlabs is een uitgave van
Elektor International Media B.V.
Postbus 11, 6114 ZG Susteren, Nederland
Tel.: +31 (0)46- 4389444,
Fax: +31 (0)46-4370161

Nieuwe abonnementen & bestellingen
service@elektor.nl Tel. 046-4389444

Elektor International Media B.V. legt gegevens vast voor de uitvoering van de (abonnements) overeenkomst. De door u verstrekte gegevens kunnen gebruikt worden om u te informeren over relevante diensten en producten. Stelt u daar geen prijs op, dan kunt u dit schriftelijk doorgeven aan:

Elektor International Media B.V.,
Afdeling Customer Service
Postbus 11, 6114 ZG Susteren.

Of per email: service@elektor.nl

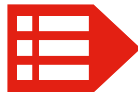
In overeenstemming met de Wet bescherming persoonsgegevens zijn de verwerkingen van persoonsgegevens aangemeld bij de toezichthouder, Autoriteit Persoonsgegevens te Den Haag.

Druk: Senefelder Misset, Doetinchem
Distributie: Betapress, Gilze

Advertenties Benelux

Margriet Debeij
Tel. +49 2 41/955 09 174
Mobiel: +49 (170) 550 539 6
E-Mail: margriet.debeij@eimworld.com

Advertentietarieven, nationaal en internationaal, op aanvraag. Alle advertentiecontracten worden afgesloten conform de Regelen voor het Advertentiewezen gedeponeerd bij de rechtbanken in Nederland. Een exemplaar van de Regelen voor het Advertentiewezen is op aanvraag kostenloos verkrijgbaar.





De spanning stijgt...

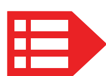
Als *editor-in-chief* worden mijn vakanties gedicteerd door de data waarop dit tijdschrift elke twee maanden naar de drukker moet — data waarover geen discussie mogelijk is. Afgelopen zomer heb ik een heleboel tijd op kantoor moeten doorbrengen met het testen van een nieuw *content management system* dat de productie van ElektorLabs moet stroomlijnen. Maar halverwege september was er wonder boven wonder een gaatje en kon ik eindelijk twee weken op vakantie. Teruggekomen uit het heerlijke Oostenrijk zei dat systeem echter dat ik nog maar één week had om het nummer dat u nu leest af te werken — als alijd tot aan de nok gevuld met artikelen niet alleen voor oude rotten in het vak maar ook voor beginners.
Ik weet zeker dat u een grote belangstelling (al dan niet beroepsmatig) hebt voor elektronica, net als mijn collega's hier bij Elektor Labs en ikzelf. En in dat geval hebt u gegarandeerd alle symptomen van de 'electronica'-koorts. Immers: de 'grootste elektronicabeurs ter wereld' die elke twee jaar wordt georganiseerd, staat te beginnen en naar verwachting worden alle bezoekers- en exposantenrecords dit jaar tussen 13 en 16 november weer gebroken. En Elektor is ook van de partij, en hoe! In hal C5 vindt u de grootste stand die we ooit hadden (C5/225). Samen met Messe München en gerenommeerde sponsors als Arrow, Avnet Silica en EBV Elektronik organiseren we daar voor de tweede keer de 'electronica Fast Forward'-competitie. Innovatieve startups en slimme solo-ontwerpers strijden op de vierde dag van de beurs om de begeerde prijzen in drie categorieën. Ook is er gezorgd voor interessante omlijsting in de vorm van lezingen en workshops — waaronder Eben Upton (oprichter van de Raspberry Pi Foundation) met wie ik een vraaggesprek zal hebben. Daarnaast is het mogelijk een praatje te maken met redacteuren en labmedewerkers van Elektor, uw eigen projecten te laten zien, een van ons aan zijn jasje te trekken, uw beklag te doen of een selectie van producten uit onze shop te bewonderen.

Tot ziens in München!

Eric Bogers

Ons team

Hoofdredacteur:	Thijs Beckers (thijs.beckers@eimworld.com)
Internationale redactie:	Jan Buiting, Jens Nickel, Mariline Thiebaut-Brodier
Technische redactie:	Ton Giesberts, Luc Lemmens, Denis Meyer, Clemens Valens, Jan Visser
Aan dit nummer werkten mee:	Jelle Aarnoudse, Rolf Blijleven, Eric Bogers, Kees de Groot, Martien Jansen, Evelien Snel
Ledenmanagement:	Raoul Morreau (raoul.morreau@eimworld.com)
Vormgeving & Layout:	Giel Dols, Mart Schroijen, Patrick Wielders
Hoofd Online:	Daniëlle Mertens
Directeur:	Don Akkermans



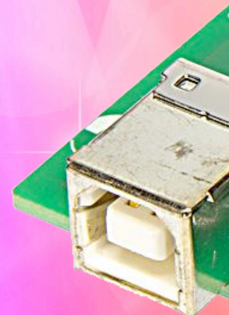
in dit nummer

58e jaargang – nr. 650

november/december 2018

Compacte USB/DMX-converter

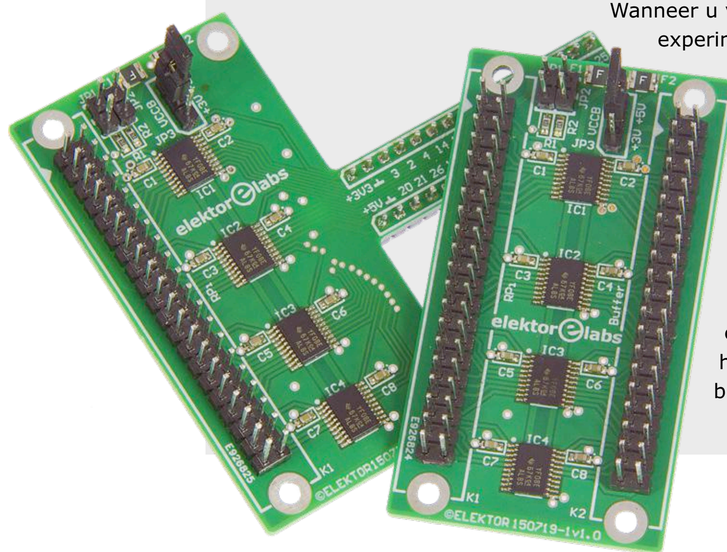
volledig elektrisch geïsoleerd

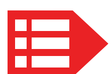


12 **Raspberry Pi bufferboard**

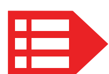
nooit meer opgeblazen I/O's

Wanneer u veel met de Raspberry Pi experimenteert en regelmatig allerlei externe hardware aansluit de GPIO's via de header, zal het u misschien al een of meerdere keren gebeurd zijn: door een te hoge spanning of statische ontlading is de poort opgeblazen en kunt u uw geliefde creditcard-computer eigenlijk weggooien. Met het hier beschreven Raspberry Pi bufferboard voorkomt u dit!





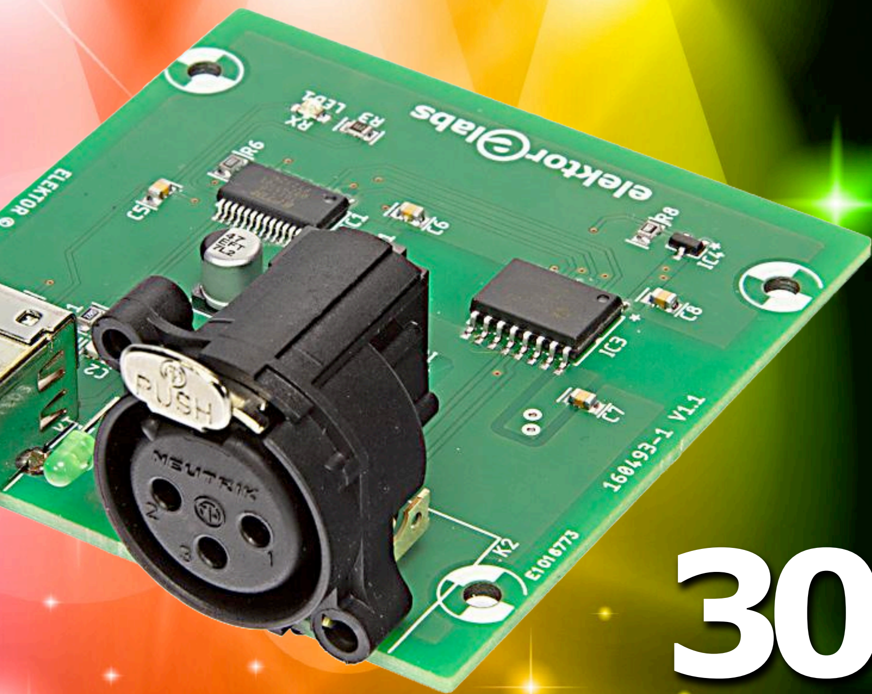
achtergrond & info



projecten

27 **Opamps in de praktijk**

deel 1 : principes



elektorlabs magazine

Hamster Run-O-Meter

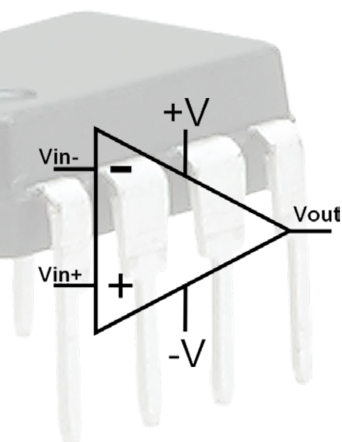
km/h en afstand meten

Bij menig knaagdier hangt er wel zoiets in de kooi: een wiel waarin ze naar hartenlust (of tot nachtelijke ergernis van het baasje) in kunnen rennen. Als elektronicus vráágt dat natuurlijk om experimenteren. Niet met het diertje (dat is de biotechnologie), maar met het bewegende wiel. In dit eenvoudige project loggen we de topsnelheid en de afgelegde weg.





79



binnenkort

Elektorlabs magazine januari/februari 2019

3-assige CNC-machine • zelfbouw-soldeerstation • RDS-FM-ontvanger voor de Raspberry Pi • weerstation-bouwpakket • cross-platform BASIC-compiler • USB-stroombegrenzing/meter • snurk-alarm voor de Arduino • functiegenerator • motorsturing: de principes • veilige communicatie met ATmega328 • styropor-snijapparaat • Android en C op de Raspberry Pi • en nog veel meer.

Aankondigingen onder voorbehoud.
Elektorlabs januari/februari 2019 verschijnt op 28 december 2018.

Elektor Business Magazine 6/2018

Elektor Business Magazine 6/2018 is onze beurs-special naar aanleiding van electronica 2018 in München (13-16 november 2018) waar dit nummer ook zal worden verspreid. Aan dit nummer is meegewerkt door bedrijven en onderzoeksinstituten, maar ook door zelfstandige auteurs; de bijdragen behandelen onder andere productie, medische elektronica, IoT en meer. Plus natuurlijk nieuwe afleveringen van de vaste EBM-rubrieken zoals Infographics en Business Store.

Elektor Business Magazine editie 6/2018 verschijnt in november 2018 in druk voor Elektor Gold-leden en als pdf-download voor Elektor Green-leden. Het nummer is ook te koop via www.elektormagazine.com. Aankondigingen onder voorbehoud.



Ultra-low-power ontwerp

tips, trucs en valkuilen

Andreas Riedenauer (Ineltek GmbH, Duitsland)

Fabrikanten van elektronische onderdelen vermelden bij veel van hun producten dat ze door hun lage energieverbruik geschikt zijn voor het IoT. Maar het energieverbruik van een toepassing wordt niet alleen bepaald door de keuze van de onderdelen, er zijn ook minder voor de hand liggende risico's en valkuilen.

Pico Power, Nanowatt Technology, Deep Sleep Mode en Zero Power Oscillator (!) zijn niet alleen wervende teksten uit commerciële brochures, u vindt ze ook in 'objectieve' datasheets. Maar wat betekent dit eigenlijk? Waar moeten we bij het ontwerpen van een energiezuinige schakeling of toepassing op letten? De meeste moderne elektronische schakelingen worden door microcontrollers (MCU's) bestuurd, die samen met de periferie (en in combinatie met de software) het energieverbruik van het apparaat bepalen.

Bron: Wilgengebroed, CC BY 2.0, https://commons.wikimedia.org/wiki/File:Internet_of_Things.jpg

Microcontrollerschakelingen

Bij de keuze van een MCU speelt de processorkern een steeds minder prominente rol. De meeste toepassingen worden waar mogelijk platform-onafhankelijk in C geprogrammeerd. Belangrijker dan een bepaalde processorkern zijn factoren zoals ontwikkeltools, verschillende uitvoeringen binnen een familie wat betreft geheugenomvang, aantal pinnen, soorten behuizingen en randapparatuur, maar ook de leverbaarheid op lange termijn en de kosten. De meeste fabrikanten hebben speciale low-power-modellen in hun assortiment, maar met het juiste ontwerp kunnen ook veel standaard-MCU's spaarzaam omgaan met energie.

Bij de keuze voor een 8-bits of een 32-bits controller is de vraag van belang of u echt 32 bits nodig hebt. Er zijn weliswaar goedkope 32-bits controllers, maar u moet deze zorgvuldig met moderne (!) 8- of 16-bits MCU's vergelijken. Vaak hebben de goedkope 32-bits MCU's eenvoudige periferie en weinig extra functies. Veel 8/16-bits controllers zijn niet alleen energiezuiniger en goedkoper, maar dankzij speciale periferie ook technisch superieur. Dan geldt: minder is meer!

Structuurgrootte

Hoe kleiner de afmetingen van de halfgeleiderelementen op de chip, des te groter de lekstromen. Een 8- of 16-bits controller in een 130-nm CMOS-proces kan zodanig worden gefabriceerd dat alle vluchtige geheugens ook in de diepste slaapmodus hun inhoud behouden, bij een verbruik dat ruim onder een halve µA blijft. Een volgens hetzelfde proces gefabriceerde 32-bits controller verbruikt door het grotere aantal elementen en de bredere bussen altijd meer energie.

Grotere afmetingen zorgen niet alleen voor kleinere lekstromen, maar verminderen ook de gevoeligheid voor EMC. Om te hoge interne veldsterktes te vermijden, werken hooggeïntegreerde MCU's met een verlaagde CPU-spanning, waardoor de stoorspanningsafstand kleiner wordt en dus de gevoeligheid voor stoorvelden wordt verhoogd. Daarnaast verbruikt ook de interne spanningsomzetter energie en heeft deze een zekere aanloopperiode.

Klok

Omdat het stroomverbruik van CMOS-logica voor een groot deel evenredig is met de klokfrequentie, is ook hier energiebesparing mogelijk. Het lijkt voor de hand liggend om de klokfrequentie zodanig te verlagen dat de gewenste taak nog net

kan worden uitgevoerd. Maar zo eenvoudig is het niet: bij de meeste toepassingen worden niet alleen berekeningen uitgevoerd maar vindt ook datatransport plaats. Dit laatste vindt meestal binnen een bepaald tijdsbestek plaats, bijvoorbeeld bij serieel transport met een voorgeschreven baudrate. In zulke gevallen kan, zoals in het **kader** is te zien, door het omschakelen van de klokfrequentie flink op energie worden bespaard. Tijdens de communicatie moet de klokfrequentie net voldoende zijn, bij berekeningen moet de CPU met de hoogst mogelijke frequentie werken en daarna in de slaapstand worden gezet. Dit spaart energie ten opzichte van een lange actieve fase bij een lagere frequentie (met dezelfde voedingsspanning).
Welke klokgenerator moeten we kiezen? Als er gebruik wordt gemaakt van slaapmodi, is de interne RC-oscillator meestal de eerste keus (bij voldoende nauwkeurigheid). In twijfelgevallen is het handig om verschillende typen (ook uit dezelfde productfamilie) te vergelijken. Moderne RC-oscillatoren hebben een afwijking van maximaal 2% over het volledige V_{CC}- en temperatuurbereik. Als de jitter van belang is kunt u de betreffende specificaties bij de fabrikant opvragen. Het is al eens voorgekomen dat nieuwe versies van hetzelfde onderdeel aanzienlijk meer jitter produceerden dan hun voorgangers, zonder dat dit expliciet in de datasheet werd vermeld. Het kan dus nuttig zijn dat oudere versies van onderdelen lang leverbaar blijven. Als de RC-oscillator niet nauwkeurig of frequentiestabiel genoeg is, komt vanwege de kortere aanlooptijd eerst een keramische resonator in aamerking, en dan pas een kristal.

Periferie
Het is handig als perifere eenheden direct met elkaar kunnen communiceren. Dit is niet alleen goed voor de energiebesparing, maar vereenvoudigt ook het ontwerp, verhoogt de prestaties, vermindert het aantal regels code en maakt een klokgestuurde reproduceerbare timing en synchronisatie tussen perifere eenheden mogelijk. Ook kan eenvoudiger aan beveiligingseisen worden voldaan. Relevante termen zijn DMA, Event System en Core Independent Peripherals (CIP). Hierdoor kan de CPU zonder onderbrekingen andere taken uitvoeren of gewoon blijven slapen.
Bij moderne low-power-MCU's is het essentieel dat periferie die even niet wordt gebruikt, kan worden uitgeschakeld. Vergelijk hiervoor de datasheets (inclusief errata) van vergelijkbare onderdelen met elkaar, en bekijk ook de betreffende application notes van de fabrikant.
Het uitschakelen gebeurt door de klok te 'bevriezen' of de voedingsspanning intern te onderbreken. Hoewel bij CMOS-logica zonder kloksignaal alleen een kleine lekstroom vloeit, loont het soms om ook de interne voedingsspanning uit te schakelen.
Als de controller over CCL (Customer Configurable Logic) beschikt, kan hiermee ook aan de energiebesparing worden bijgedragen doordat bepaalde functies sneller en zuiniger in hardware kunnen worden uitgevoerd.

Ingangspennen
Open ingangen (Floating Inputs) zijn natuurlijk uit den boze, maar het aansluiten op logische '0'- of '1'-niveaus garandeert niet het laagste stroomverbruik. Hiervoor mogen de ingangsspanningen niet meer dan 0,5 V van GND of V_{CC} afwijken.

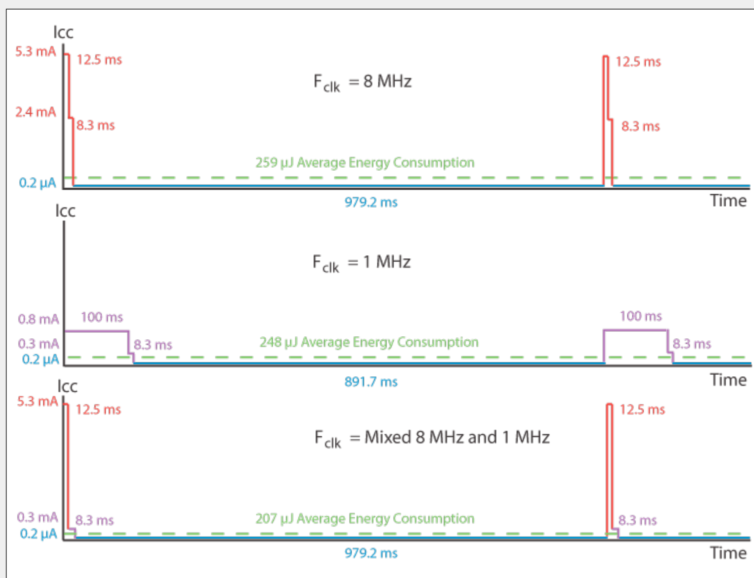
Real Time Clock
Veel microcontrollers beschikken over een interne RTC waar alleen een klokkristal op hoeft te worden aangesloten. Dit is weliswaar goedkoper dan een externe RTC, maar heeft ook nadelen. Het stroomverbruik is aanzienlijk hoger, en de nauwkeurigheid lager dan bij (met name temperatuurgecompenseerde) externe modules. De zuinigste RTC's met geïntegreerd kristal verbruiken zo'n 40 nA, met temperatuurcompensatie is dit ongeveer 100 nA. Sommige hebben een extra RC-oscillator met lagere nauwkeurigheid, die echter maar 17 nA verbruikt. Modules met geïntegreerd kristal zijn altijd nauwkeuriger dan de verouderde versies met extern kristal. Hoewel deze laatste voor afwijkingen tot 5 ppm zijn gespecificeerd, worden ze door het solderen aanzienlijk minder nauwkeurig. Dit kan bij geïntegreerde modules en het juiste soldeertemperatuurprofiel niet gebeuren. De schakeling en de layout moeten bij externe kristallen extra zorgvuldig worden ontworpen, anders kunnen er grotere frequentieafwijkingen en zelfs opstartproblemen ontstaan, in het bijzonder bij oscillatoren met een hoge impedantie. Professionele kristalfabrikanten kennen de karakteristieken van verschillende MCU's en kunnen helpen bij het ontwerp. Probeer vooral niet zelf met uw standaard meetprobe iets aan een klokkristal te meten!

Iets over 'Typical Values'
Vergeet de in de datasheets opgegeven 'Typical Values'! Deze waarden zijn geschikt voor promotionele doeleinden, maar meestal nutteloos voor serieproducten waarbij van worst-case-scenario's moet worden uitgegaan, en dus van het maximale stroomverbruik dat vaak hoger is dan de typische waarde. 'xx µA per MHz' betekent niet het verbruik van de microcontroller bij exact één megahertz! Die is in het algemeen veel hoger. De juiste informatie krijgt u door de datasheet inclusief de voetnoten nauwkeurig te bestuderen.

Geheugens
De meeste MCU's hebben flashgeheugens die via bootloaders kunnen worden bijgewerkt. Als u hier incidenteel kleine hoeveelheden data in wilt opslaan, controleer dan of de pagina's van het geheugen in uw controller klein genoeg zijn. Anders moeten er steeds onnodig veel geheugencellen worden overschreven, en dat kost energie, schrijftijd (nog meer energie) en levensduur. EEPROM's kunnen tien tot honderd maal vaker worden overschreven dan (NOR-)Flash. Externe EEPROM's kunnen dus een zinvol alternatief zijn, met name bij MCU's zonder intern EEPROM. Conductive Bridging RAM (CBRAM) is een moderne technologie die voor het schrijven 25 tot 50 keer minder energie nodig heeft dan EEPROM, in combinatie met een zeer korte schrijftijd, stralingsimmuniteit, lage bedrijfsspanning en geringe kosten. Het gegarandeerde aantal schrijfcycli ligt op minstens 100 K en is anders dan bij flash en EEPROM temperatuuronafhankelijk. Ook het databehoud mag er wezen: zelfs in continubedrijf bij 150 °C wordt 10 jaar gehaald. De maximale geheugengrootte is op dit moment 512 Kbit, grotere modellen zijn in ontwikkeling.

Condensatoren
Ontkoppelcondensatoren zijn onmisbaar in elektronische schakelingen met MCU's, FPGA's en dergelijke. Vaak is niet alleen ontkoppeling van gedeelten van schakelingen nodig maar ook het bufferen van energie. Standaardcondensatoren voor low-power-toepassingen zijn vaak niet geschikt voor ultra-low-power-apparaten met een zeer lange batterij-levensduur. Het is

Dynamisch aanpassen van de klokfrequentie



Voor deze toepassing werd een klassieke ATmega128 gebruikt omdat pico-power AVR's met 128 K flash nog niet beschikbaar waren. De sleutelzender bevindt zich gedurende 98% van de tijd in power down-modus en wordt voor testdoeleinden één keer per seconde met een druk op een knop geactiveerd. De voedingsspanning is 3 V. De communicatie met het basisstation gebeurt draadloos en versleuteld met AES met een datarate van 9600 baud. Voor de dataverwerking inclusief encryptie zijn ongeveer 100.000 klokcycli benodigd, per uitzending worden 8 bytes verstuurd.
Eerst wordt de maximale klokfrequentie van 8 MHz van de RC-oscillator gebruikt. Er is dan een totale energiebehoefte van 250 µJ per uitzending. Wordt de klokfrequentie gereduceerd tot 1 MHz, dan kan tijdens de door de baudrate bepaalde duur van de communicatie duidelijk energie worden bespaard. De wiskundige operaties worden dan eveneens met verminderde stroomopname uitgevoerd, maar duren dan langer. MHz, dan komen we op slechts 207 µJ, een energiebesparing van 20%!

algemeen bekend dat elektrolytische en tantaalcondensatoren een grote lekstroom hebben, maar wist u dat deze ook bij keramische multilayer-condensatoren (MLCC) wel 5 µA kan bedragen?

Voedingsspanning

In energiezuinige apparaten worden bij voorkeur lage spanningen (bijvoorbeeld 5 V) gebruikt. Meestal is de spanningsbron – batterij of accu – bepalend voor de randvoorwaarden. Over het onderwerp 'keuze van de juiste energiebron' kan een compleet artikel worden geschreven; hier volgen in ieder geval een paar tips.
De zelfontladingsstroom van kleine alkalinebatterijen bedraagt bij kamertemperatuur 5...10 µA. In dat geval is het niet nodig om bij het ontwerp veel moeite te doen om een paar honderd nA te sparen. Batterijfabrikanten zijn vaak terughoudend met relevante documenten; de meeste kans maakt u bij fabrikanten van bekende merken.
De typische 1,5 V van een alkalinebatterij is te weinig voor de meeste microcontrollers, vooral omdat de spanning tijdens bedrijf nog daalt. Als de beschikbare ruimte het toelaat, kunnen twee cellen in serie worden gebruikt. Een alternatief zijn DC/DC-boostconverters die ruimte besparen en de spanning zo lang mogelijk op een constante waarde van bijvoorbeeld 3 V stabiliseren, zelfs bij een ingangsspanning van 0,7 V of nog minder. Bij een zorgvuldig ontworpen schakeling worden de verliezen van moderne converters ruimschoots gecompenseerd door het efficiëntere batterijgebruik. Als er een mechanische aan/uit-schakelaar is, worden aan de converters geen extra eisen gesteld. Moet het systeem daarentegen worden geactiveerd door een signaal op een I/O-pen, een timer, een temperatuurverandering (uitzondering: mechanische thermostaat), een geluid of een radiosignaal, dan moet de controller of een ander deel van de schakeling continu in bedrijf blijven. In dat geval moet de DC/DC-converter ook bij geringe belasting een extreem laag eigen verbruik hebben.
Het is dan beter als het sensordeel van de schakeling op een veel lagere spanning werkt dan de controller, en tevens het laagste eigen verbruik heeft (een zeldzame combinatie!), want dan kan de converter tijdens de slaapmodus worden uitgeschakeld. Nog beter is het om een van de weinige ultra-low-power-microcontrollers te gebruiken die aan 1,2 V of minder genoeg hebben. Ze zijn er sinds kort ook met Cortex M0+-cores, en typen voor minder dan 1 V zijn in ontwikkeling.
Als het te ontwikkelen apparaat sowieso een hogere voedingsspanning nodig heeft dan de controller, kunt u uw creativiteit de vrije loop laten. Een voorbeeld: vier batterijen in serie moeten zowel de elektronica als een motor, magneten en relais voeden. De spanning van deze serieschakeing (6 V) daalt in de loop van de tijd tot 3,6 V. In dit geval kunnen we de elektronica met het midden van de serieschakeling verbinden. De spanning daalt van 3 V naar 1,8 V, maar blijft binnen het bereik van veel standaardtypes. DC/DC-converters of LDO's zijn hier overbodig.
De inwendige weerstand van lithium-knoopcellen is meestal enkele tientallen ohm en stijgt tijdens het ontladen tot meer dan 100 Ω. Dat is teveel voor grote impulsbelastingen. Bij accu's (en zeker bij LiPo-cellen) moet diepe ontlading worden vermeden.

Radioverbindingen

Draadloze netwerken zijn veel ouder dan het IoT, denk maar aan hoorapparaten of DCF77-klokken. Ook wearables bestaan al tientallen jaren. Nochtans vormen sport- en medische producten, loggers, gebouwautomatisering en verlichting snel groeiende markten op dit gebied.

Long Range Radio

De energiebehoefte van een radioverbinding hangt af van het zendvermogen, de datarate, de bandbreedte, de gevoeligheid van de ontvanger en de antenne. Met low-power zenders in het sub-GHz HF-gebied is bij kortdurende uitzendingen met lage datarates een reikwijdte van 5...30 km haalbaar. Onder deze omstandigheden is jarenlang bedrijf met een batterij mogelijk. Populaire standaarden zoals LoRa en Sigfox worden vaak gebruikt voor toepassingen zoals een inbraakalarm in afgelegen gebouwen, diefstal of vandalisme, procesbesturing, meteorologie en het verzamelen van meetwaarden, tracken van dieren en nog veel meer.

Met WSPR (Weak Signal Propagation Reporter) op de kortegolf wordt met een zendvermogen van 10 mW een afstand van 1000 km overbrugd, en met 100 mW zelfs 20.000 km – de halve wereld rond! De datarate is met 1 bit/s weliswaar erg laag, maar voldoende voor bijvoorbeeld het dagelijkse weerbericht vanaf de Zuidpool.

Short Range Radio

Keyless entry, transponders en sommige fietscomputers zijn typische draadloze toepassingen voor korte afstanden in de langegolfband. Voor deze lage frequenties bestaan extreem zuinige ontvangers die ongeveer 2 μA verbruiken bij continubedrijf, dus niet alleen tijdens korte inschakelperiodes. Aan de bovenkant van het radiospectrum, in het UHF/GHz-gebied, vinden we de bekende standaarden WiFi, BTLE en Zigbee.

Voor keyless entry worden beide frequentiegebieden gebruikt. De langegolf wordt gebruikt om een speciale chip in de autosleutel te activeren als deze in de buurt van de auto komt. De chip reageert op een bepaald modulatiepatroon van het langegolfsignaal van de auto en activeert op zijn beurt de UHF-transceiver, die de eigenlijke bidirectionele versleutelde communicatie met de auto voor zijn rekening neemt.

Er is ook nog een andere interessante combinatie van dit soort onderdelen. Hierbij demoduleert een diodeontvanger een hoogfrequent radiosignaal met dubbele modulatie. Valt het signaal binnen de frequentieband van de 'wake-up-receiver' dan wordt het hieraan doorgegeven. Is ook het patroon correct, dan activeert de wake-up-receiver de rest van de schakeling waarna de data-uitwisseling kan plaatsvinden. Ondanks dat de ontvanger onafgebroken klaarstaat voor ontvangst, bedraagt de standby-stroom maar zo'n 2 μA. Dit is ongeveer 1 promille van de gebruikelijke stroomopname van een BTLE-module! Met de juiste uitbreiding van het protocol zou deze procedure ook bij BTLE, WiFi en andere standaarden kunnen worden toegepast.

NFC

Near Field Communication (NFC) is ook interessant voor ultra-energiezuinige producten. Passieve NFC-tags bestaan al lang. Deze kunnen net als andere tags (transponders) worden gebruikt, maar kunnen ook door smartphones met NFC-lezer worden uitgelezen. Sommige van deze tags bevatten encryptietechniek en zijn fraudebestendig gebouwd om illegaal kopiëren (anti-cloning) en andere ongeoorloofde handelingen tegen te gaan.

Een NFC-interfacechip geeft willekeurige apparaten de mogelijkheid om via NFC te communiceren (bijvoorbeeld voor een service-interface). Deze chips zijn goedkoop en kunnen eenvoudig zonder speciale HF-expertise en, omdat ze passief zijn, zonder vergunning worden geïmplementeerd.

De chip, indien nodig plus de aangesloten controller, wordt gevoed door een geïntegreerde energy harvesting-module. De energie die aan het NFC-veld van de lezer wordt onttrokken is voldoende om een microcontroller via de bootloader te programmeren. Veel van deze onderdelen bevatten daarnaast een eigen EEPROM. Er zijn ook microcontrollers met geïntegreerde NFC-interface.

Energy-harvesting is ideaal voor IoT-toepassingen, waarbij energiebesparing vrijwel altijd van belang is. Een sensor die incidenteel wordt gebruikt, kan als hij geen loggerfunctie heeft zonder batterij worden gerealiseerd. Zo kan bijvoorbeeld een vochtigheidssensor die in een muur of vloer is ingebouwd op ieder moment, en ook nog na tientallen jaren, met een smartphone of leesapparaat worden uitgelezen. Kleine elektronische labels met e-paper-display worden bij het beschrijven via het NFC-veld gevoed en hebben daarom ook geen batterij nodig.

Actieve low-power-NFC-sensortags slaan de gemeten waarden van temperatuur, versnelling, vochtigheid, straling en andere grootheden vaak jarenlang op in niet-vluchtige geheugens, en voorzien de waarden van tijdstempels die van de geïntegreerde RTC afkomstig zijn. Zulke tags bevatten in één chip de meeste benodigde onderdelen, dus naast de NFC-interface een RTC, een EEPROM, een ADC, een temperatuursensor en een analoge interface voor externe sensoren. ◄

(180440-03)

Links and literature

[1] D. Oshinubi, „Energieeffiziente Auswertelogik für kapazitive mikro-mechanische Drehratensensoren", KIT Scientific Publishing, Karlsruhe 2010; ISBN 978-3-86644-449-2

[2] Horowitz/Hill, "Elektronica Kunst en Kunde, deel 2", Uitgeverij Segment BV, 1996

[3] Klasche, Hofer, Pelka, Nührmann, „Professionelle Schaltungstechnik, Band 2", Franzis Verlag, München 1993

[4] R. Sarpeshkar, „Ultra Low Power Bioelectronics", Cambridge, 3rd edition 2013; ISBN 978-0-521-85727-7

[5] Keithley, „Low Level Measurements Handbook", 6th edition, Free download: www.keithley.com

[6] Epson, „New Series for Low Power Application", Training Paper

[7] J. Ganssle, „Hardware and Firmware Issues in Using Ultra-Low Power MCUs", Rev. 4, March 2016

[8] A.M. Holberg, A. Saetre, „Innovative Techniques for Extremely Low Power Consumption with 8-bit Microcontrollers", Atmel White Paper

[9] A. Riedenauer, 'Iedere microwatt telt', Elektor Magazine 3/2010: www.elektormagazine.nl/090157



(Bijna) alles wat u altijd al wilde weten over...

energy harvesting

met antwoorden van **Robert Lacoste**



foto : shutterstock.com

V *Het oogsten van energie (Engels: energy harvesting) staat erg in de belangstelling. Wat wordt precies met deze term bedoeld?*

A Ik weet niet zeker of er een officiële definitie bestaat, maar in de praktijk worden hiermee alle technieken bedoeld die het mogelijk maken om een klein elektronisch systeem te voeden met omgevingsenergie. Dat wil zeggen uit energiebronnen die 'toevallig' aanwezig zijn rond het object en voldoende energie kunnen leveren om het te laten werken.

V *Over welke energiebronnen hebben we het hier?*

A In de eerste plaats uiteraard de zon. Het 'oogsten' hiervan is niet nieuw: rekenmachines op zonne-energie bestaan al sinds het einde van de jaren zeventig... Maar er komen nog veel meer bronnen in aanmerking: warmte, mechanische bewegingen, trillingen, vloeistofstromen, elektrochemische reacties, elektromagnetische straling, enzovoort. We worden omringd door energie, het probleem is alleen hoe je deze kunt benutten...

V *Stel dat ik een elektronisch product wil ontwerpen dat wordt gevoed met geoogste energie. Wat is dan de eerste stap?*

A In het algemeen kunnen energiebronnen uit de omgeving heel weinig energie leveren (met uitzondering van grote zonnepanelen op het dak...). Het ligt dus voor de hand dat u moet beginnen met het beperken van het energieverbruik door de elektronische schakeling van het product te optimaliseren. Hierbij telt iedere microampère... Elders in dit nummer kunt u daar meer over lezen.
Bij het elektronisch ontwerp moet ook rekening worden gehouden met een gemeenschappelijke eigenschap van veel energiebronnen uit de omgeving: ze zijn vaak onberekenbaar. Stel dat u een fiets-tracker hebt ontworpen die wordt gevoed door de rotatie van een van de wielen. Als de fiets wordt gestolen en in een busje wordt vervoerd heeft deze tracker weinig zin. Dus moet er reserve-energie aanwezig zijn (accu of supercondensator), of een alternatieve energiebron (batterij). Hetzelfde geldt bijvoorbeeld voor een thermometer die wordt gevoed door elektromagnetische velden in de omgeving. U wilt niet dat deze alleen metingen uitvoert als het WiFi-signaal van de buurman actief is. In dit geval is het misschien slimmer om de meetwaarde meer of minder vaak te laten actualiseren als functie van de beschikbare energie. Zo'n adaptieve werking is erg handig bij het gebruik van fluctuerende energiebronnen.

V *En dan??*

R De tweede stap is het schatten van de hoeveelheid energie die met harvesting kan worden verkregen. Dit is meestal niet eenvoudig, maar het is noodzakelijk om snel een orde van grootte te schatten om te kunnen beoordelen of het project haalbaar is of niet. Als uw project 1 mA verbruikt en de bron soms maar 1 µA levert, dan kunt u beter naar iets anders uitkijken! In dit stadium is een haalbaarheidsonderzoek erg nuttig.

V *Hoe weten we bij gebruik van een zonnepaneel hoeveel energie er beschikbaar zal zijn?*

A Om te beginnen zijn er twee totaal verschillende situaties mogelijk: panelen binnenshuis en panelen buitenshuis. De zonnestraling buiten hangt sterk af van de geografische positie en het seizoen. Het goede nieuws is dat er uitstekende websites zijn die onder andere informatie geven over de gemiddelde zonnestraling voor iedere maand van het jaar op een bepaalde plaats in Europa [1]. Met deze informatie en een beetje moeite kunt u te weten komen hoeveel Wh/m² er gedurende de wintermaanden per dag beschikbaar is. Op andere sites kunt u zelfs vinden hoe groot de kans is dat de zonnestraling op een bepaalde dag onder een bepaalde grenswaarde daalt. Hiermee kan de capaciteit van een buffer-accu worden berekend [2].
Veronderstel dat uw elektronische schakeling gemiddeld 1 mA bij 3 V nodig heeft, dus per dag 72 mWh, dat u in Parijs woont

en dat u een bufferbatterij wilt hebben waarmee u zich een paar dagen geen zorgen hoeft te maken over het gemiddelde verbruik. Uit de informatie op de bovengenoemde websites blijkt dat de zonnestraling in Parijs varieert van 6000 Wh/m²/dag in juni tot 793 Wh/m²/dag in december. U moet die laatste (laagste) waarde voor uw ontwerp gebruiken. Omdat zonnepanelen altijd zijn gespecificeerd voor 1000 W/m², moet u een paneel vinden dat goed is voor minstens 72 mW.h × 1000 W/m² / 793 W.h/m², dat is dus 90 mW. Vervolgens neemt u een veiligheidsmarge door het resultaat met drie of vier te vermenigvuldigen... Volgens de catalogi komt u dan uit bij een paneel van zo'n 10 x 10 cm, het project is dus uitvoerbaar.

Nu moet u de elektronische schakeling nog zodanig ontwerpen dat het paneel zo goed mogelijk wordt gebruikt. Hierbij komen MPPT-onderdelen (*maximum power point tracking*) goed van pas. Kijk bijvoorbeeld eens naar het power-management-IC ADP5091 van Analog Devices.

De MPPT-techniek wordt toegepast bij niet-lineaire elektriciteitsopwekking, bijvoorbeeld met zonne-of windenergie. De hierbij geleverde spanning wordt voortdurend aangepast om het maximale vermogen aan de belasting (accu) over te dragen.

V *En de zonnepanelen voor binnenshuis?*

A Dat is een heel ander verhaal, want het niveau van de zonnestraling heeft hier weinig invloed. Voor gebruik binnenshuis wordt een ander type zonnecel gebruikt, en de geleverde vermogens zijn erg laag. Met een hoog-rendementspaneel van 5 x 2 cm (Sanyo AM-1805) kunt u binnenshuis in een matig verlichte kamer (50 lux) zo'n 4 µA bij 2,6 V verwachten.

V *Hoe kunnen we energie oogsten uit een object dat beweegt of trilt?*

A Daar zijn drie fundamenteel verschillende oplossingen voor. De eerste is het gebruik van een elektromagnetische converter, oftewel het laten bewegen van een magneet in een spoel. Een goed voorbeeld hiervan is de fietsdynamo. Er bestaan schakelaars die op dit principe zijn gebaseerd, waarbij de energie van het op de knop drukken wordt gebruikt om draadloos informatie te verzenden. De tweede is het gebruik van het piëzoelektrisch effect: druk op een stukje kwarts en er wordt energie opgewekt. Dit is het principe van de gasaansteker. Hierbij is het probleem dat het piëzoeffect een erg kleine stroom opwekt bij een hoge spanning, en daarom zijn er speciale onderdelen nodig (zoals de 'nanopower energy harvesting power supply' LTC3588-2). Tot slot kunt u gebruik maken van het elektrostatisch effect: het periodiek laten varieren van de ruimte tussen de twee elektroden van een geladen condensator wekt een geringe hoeveelheid energie op.

V *En wat betreft warmte?*

A Hierbij kunt u gebruik maken van het Seebeck- of thermoelektrisch effect (het omgekeerde van het Peltier-effect). Een temperatuurverschil tussen de twee zijden van een Peltier-module wekt een elektrische spanning op. Let op, dit moet een flink temperatuurverschil zijn: u hebt een koudebron en een warmtebron nodig. Het rendement en de uitgangsspanning zijn direct gerelateerd aan het temperatuurverschil. Met een verschil van een paar graden krijgt u bij de standaardmodules niet meer dan enkele tientallen millivolt. Er zijn dus gespecialiseerde elektronische schakelingen nodig om ze te kunnen gebruiken. Gelukkig hebben de IC-fabrikanten deze in het assortiment, zoals de LTC3109, een DC/DC-converter die al start bij een spanning van slechts 30 mV...

V *Elektor heeft enige tijd geleden een leeslamp gepubliceerd die werkt op de warmte van een waxinelicht [3]. Dit is een mooi voorbeeld van geoogste energie nietwaar?*

A Het is een prachtig project, maar eerlijk gezegd voldoet het niet aan de definitie van het oogsten van energie, omdat het waxinelicht hier met opzet is geplaatst: het is dus geen energiebron die 'toevallig' in de omgeving aanwezig is! Daarentegen is dit wel een erg mooi voorbeeld van energie-optimalisatie, omdat de thermische energie van de vlam van het waxinelicht, die normaal verloren gaat, wordt gebruikt om licht te creëren en zo het natuurlijke kaarslicht te versterken.

V *Maken deze ontwikkelingen batterijen en accu's overbodig?*

A Daar kunnen we kort over zijn, het willekeurige karakter van de energiebronnen in de omgeving vereist vaak de aanwezigheid van een kleine aanvullende batterij of tenminste een opslagmogelijkheid voor energie. Er zullen zeker toepassingen zijn waarbij alleen het oogsten voldoende is, maar daarvoor moeten we een bron hebben die 'beheersbaar' is. Dit is in het algemeen het geval bij industriële toepassingen, maar komt bij huishoudelijke producten minder vaak voor.

De volgende grappige situatie ben ik in de praktijk al een paar keer tegengekomen: ontwikkelaars hebben een productidee met daarbij de gedachte "leuk, we gaan energie oogsten!" Na enig rekenwerk komen ze erachter dat er erg weinig energie beschikbaar is, laten we zeggen enkele µW. Ze steken vervolgens veel tijd in het optimaliseren van het energieverbruik van het product, en eindigen, verheugd, met een schakeling die maar een paar µA nodig heeft. Omdat de kans bestaat dat de bron soms niet aanwezig is besluiten ze om ook nog een kleine knoopcel toe te voegen. Om vervolgens tot hun verbijstering te constateren dat ze het stroomverbruik zo goed hebben weten te beperken dat deze kleine knoopcel in z'n eentje voldoende is om het hele project jarenlang te voeden!! ◄

(180441-04)



Weblinks

[1] Fotovoltaïsch geografisch informatiesysteem: http://re.jrc.ec.europa.eu/pvg_tools/en/tools.html

[2] Europese database van daglicht en zonnestraling: www.satel-light.com/indexeS.htm

[3] "Candle2Light, lumineuze rendementsbooster", Elektor september/oktober 2017: www.elektormagazine.nl/160441

Raspberry Pi bufferboard

nooit meer opgeblazen I/O's

PROJECT-INFO



Raspberry Pi
HAT
T-Board



beginners
➡ gevorderden
experts



30 min.



SMD-soldeergereedschap



Ca. €35

Guy Weiler

Wanneer u veel met de Raspberry Pi experimenteert en regelmatig allerlei externe hardware aansluit de GPIO's via de header, zal het u misschien al een of meerdere keren gebeurd zijn: door een te hoge spanning of statische ontlading is de poort opgeblazen en kunt u uw geliefde creditcard-computer eigenlijk weggooien. Met het hier beschreven Raspberry Pi bufferboard voorkomt u dit!

Studenten zijn moordenaars! Na de pijnlijke dood van drie Raspberry Pi's besloot de auteur een kortsluitbestendig bufferboard te ontwerpen. Met een beetje googelen kwam hij op de pagina van Brian Dorey terecht [1]. De daar genoemde TXS0108E van Texas Instruments klonk interessant en het ontwerp was eenvoudig. Doordat de Raspberry Pi meerdere GND-pennen heeft, was het voor de auteur mogelijk om in totaal slechts 20 pinnen te gebruiken en zo wat ruimte vrij te laten op het breadboard. De opzet van de auteur [2] werkte prima met 1-wire, I²C en EIA232 — op 3,3 V; op 5 V kwam er wat veel ruis op de signalen.

Wij hebben de schakeling onder de loep genomen en in het lab de puntjes op de i gezet [3].

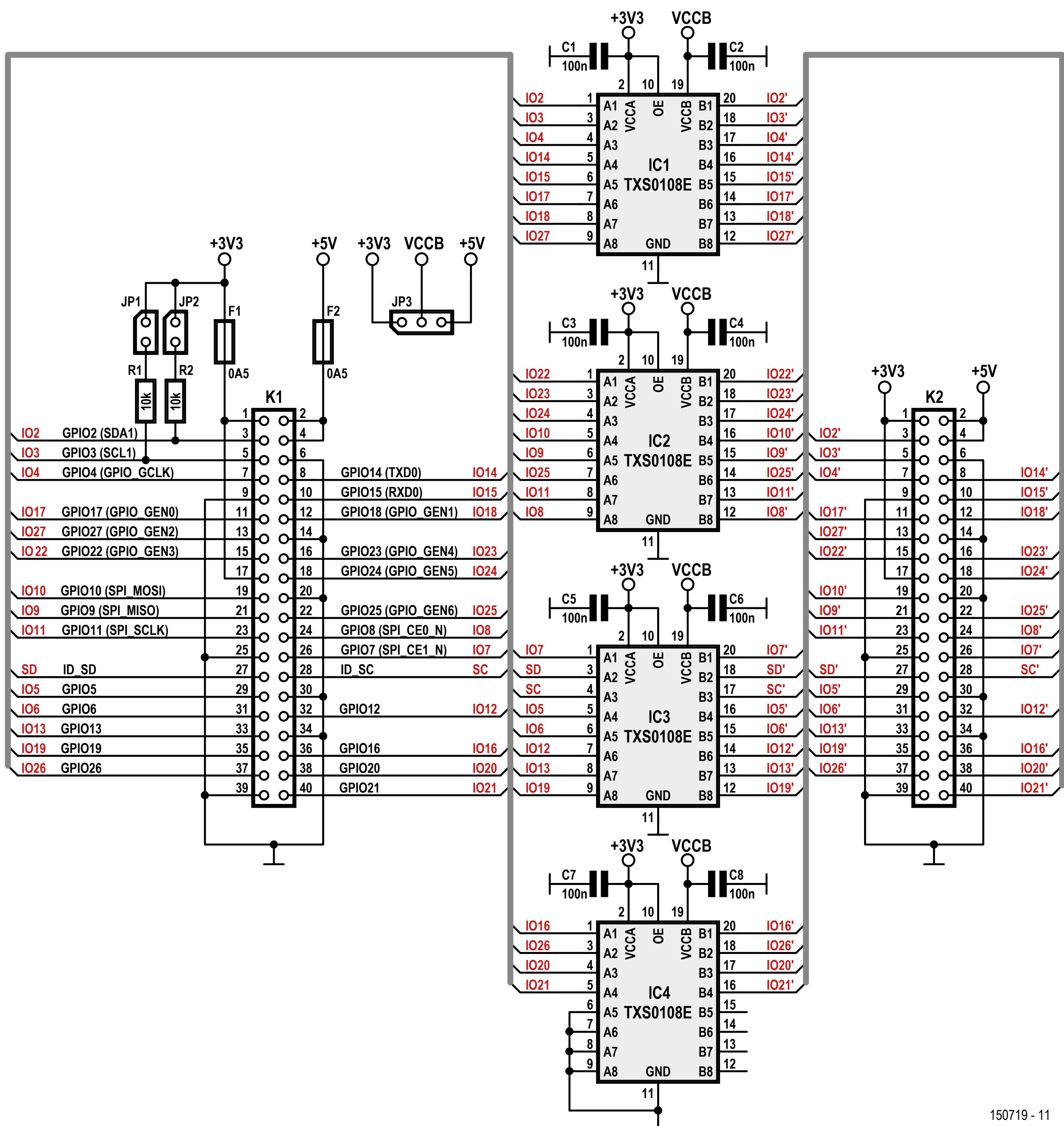
De schakeling

We hebben van deze schakeling twee versies ontworpen: de ene schakeling dupliceert de Raspberry Pi-header en houdt dezelfde pen-layout aan (**figuur 1**, print 150719-1), de andere biedt dezelfde aansluitingen op een zo compact mogelijke en breadboard-vriendelijke manier aan (**figuur 2**, print 150719-2). Deze laatste is opgezet als 'T-board' om zoveel mogelijk plaats op het breadboard vrij te laten (**figuur 3**).

De hele schakeling is in beide gevallen opgezet rond vier TXS0108E-IC's. De TXS0108E is een 8-bit bidirectionele levelshifter die ontworpen is voor open-drain- en push-pull-toepassingen. Hij ondersteunt datasnelheden tot 60 Mbps en heeft ingebouwde ESD-bescherming.

Eigenschappen

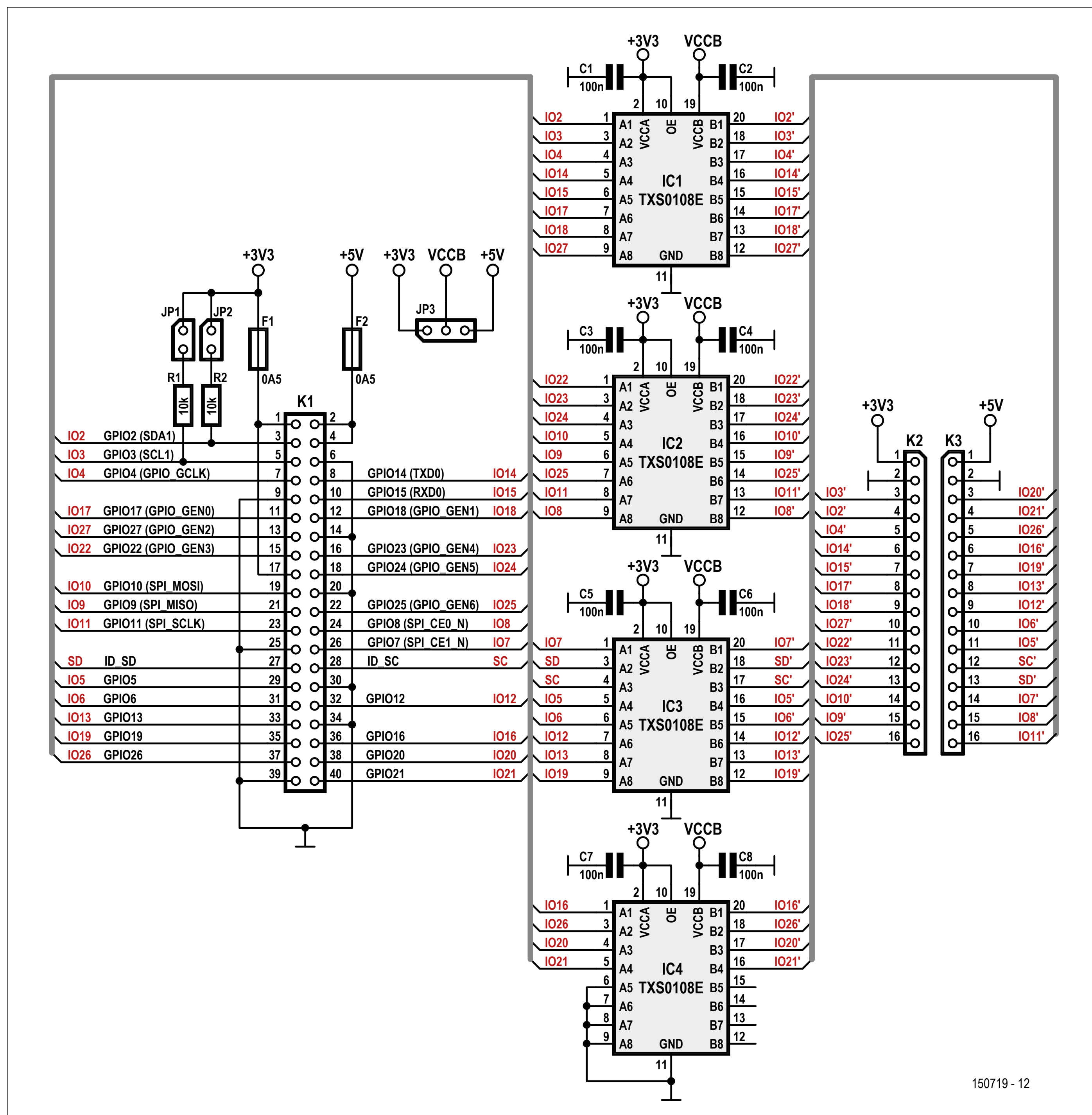
- Volledige bidirectionele buffering van alle Raspberry Pi GPIO's
- Geschikt voor 1,2 tot 5,5 V (zie tekst)
- T-board-versie zeer geschikt voor breadboard-toepassing
- ESD-veilig



Figuur 1. De schakeling is opgebouwd rond vier 8-bit bidirectionele levelshifters.

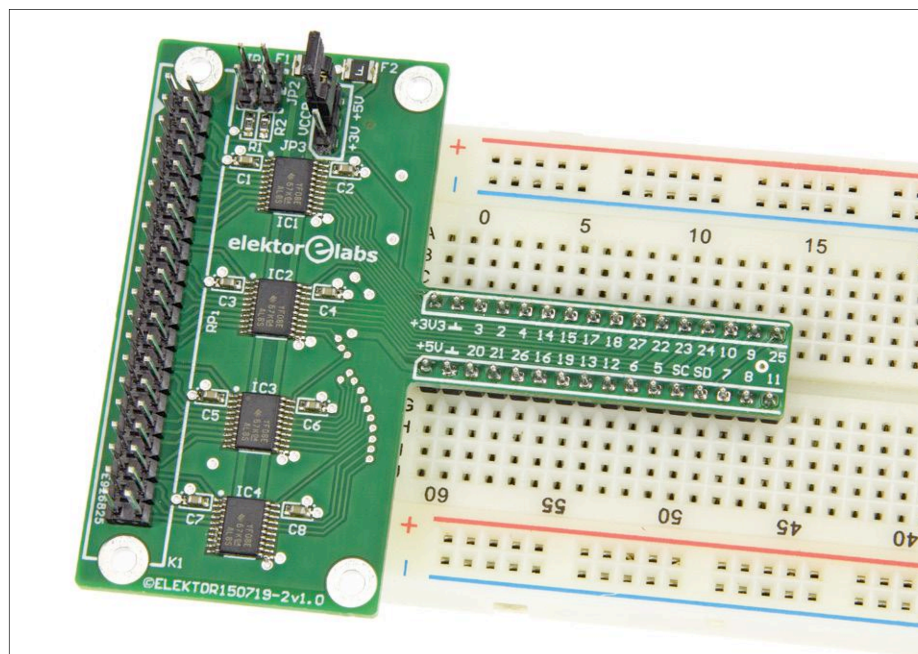
Alle I/O's beschermd – ESD-veilig

Het IC kan signalen tussen 1,2 en 3,6 V op poort A en tussen 1,65 en 5,5 V op poort B omzetten.
Niet alleen alle 26 GPIO's van de Raspberry Pi zijn gebufferd, maar ook SD en SC. Deze twee signalen worden gebruikt om een EEPROM te adresseren voor de identificatie van Pi Hats die aangesloten worden.
De pull-down-weerstanden van de GPIO's van de Raspberry Pi zijn (indien ingesteld) eigenlijk nutteloos en wellicht zelfs contraproductief wanneer het bufferboard is aangesloten. De TXS0108E heeft namelijk intern aan elke in/uitgang van de A-poort een pull-up-weerstand naar de VCCA-pen en elke B-poort-in/uitgang heeft een pull-up-weerstand naar de VCCB-pen. Deze pull-ups hebben een waarde van 40 kΩ als de uitgang laag is en 4 kΩ als de uitgang hoog is. De uitgangen van de buffers zijn dus in feite 'open drain'. Als er bijvoorbeeld een LED wordt aangesloten tussen een uitgang van de buffer en massa, ontstaat er een spanningsdeler als er een serieweerstand wordt gebruikt. Een belasting op de uit-



Figuur 2. De T-Board-versie biedt dezelfde aansluitingen bij een kleinere footprint.

gang zorgt er dus voor dat het logische hoge niveau niet meer 3,3 of 5 V is, maar lager. Iets om in gedachten te houden! Verder hebben we voor de nodige voedingsontkoppeling gezorgd met C1...C8. Beide printplaten zijn tevens voorzien van twee polyfuses (F1, F2) om de +5V- en +3,3V-voeding van de Raspberry Pi te beschermen. Daarnaast kunnen op de I²C-lijnen extra pull-up-weerstanden van 10k (R1 en R2) worden ingeschakeld door jumpers JP1 en JP2 te zetten. En als laatste kunnen de B-poort I/O's van de IC's (de gebufferde poorten dus) met jumper JP3 ingesteld worden op +3,3V- of +5V-logica.



Figuur 3. Door de print een T-vorm mee te geven, kan het breadboard vrijwel optimaal benut worden.

Aansluiten

Als K1 een gewone 2x20-pinheader is, kan het aansluiten van de Raspberry Pi-kant van de print gedaan worden met een korte 40-aderige bandkabel met twee 2x20-connectoren. Het is ook mogelijk om een extra hoge vrouwelijke 2x20-*stacking header* voor K1 te gebruiken en de print direct op de Raspberry Pi te plaatsen, zoals de meeste HATs doen. In het geval van het T-Board is dit misschien niet zo'n goed idee. Het breadboard moet dan namelijk precies op de juiste hoogte geplaatst worden. Het voordeel zou zijn dat je dan wel nog steeds een HAT op de Raspberry Pi zou kunnen plaatsen.

De uitgang van print 150719-1 kan ook met een korte 40-weg bandkabel met twee 2x20-connectoren of eraan gesol-



ONDERDELENLIJST

PCB 150719-1

Weerstanden

R1,R2 = 10 k, SMD 0603

Condensatoren

C1...C8 = 100 n, 50 V, 10 %, X7R, SMD 0603

Halfgeleiders

IC1...IC4 = TXS0108EPWR, SMD TSSOP-20

Diversen

K1 = 2x20-pens header, verticaal, steek 2,54 mm of 40-pens GPIO stacking header: 2x20, extra hoog
K2 = 2x20-pens header, verticaal, steek 2,54 mm
JP1,JP2 = 2-pens header, verticaal, steek 2,54 mm
JP3 = 3-pens header, verticaal, steek 2,54 mm
JP1,JP2,JP3 = Shunt jumper, steek 2,54 mm
F1,F2 = PPTC Resettable Fuse, smd, polyfuse, 1210L050YR Littelfuse
PCB 150719-1 in Elektor store [6]

PCB 150719-2 (T-board versie)

Weerstanden

R1,R2 = 10 k, SMD 0603

Condensatoren

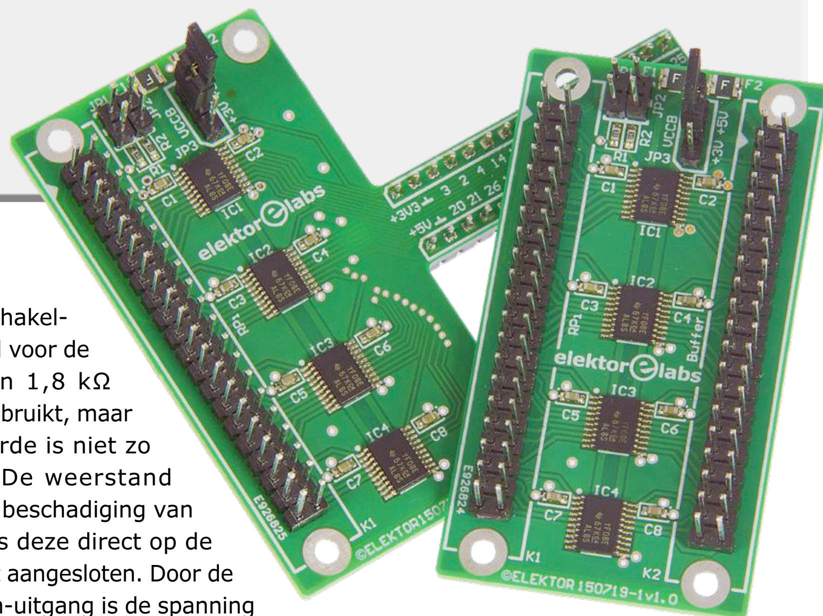
C1...C8 = 100 n, 50 V, 10 %, X7R, SMD 0603

Halfgeleiders

IC1...IC4 = TXS0108EPWR, SMD TSSOP-20

Diversen

K1 = 2x20-pens header, verticaal, steek 2,54 mm of 40-pens GPIO stacking header: 2x20, extra hoog
K2 = 2x20-pens header, verticaal, steek 2,54 mm
JP1,JP2 = 2-pens header, verticaal, steek 2,54 mm
JP3 = 3-pens header, verticaal, steek 2,54 mm
JP1,JP2,JP3 = Shunt jumper, steek 2,54 mm
F1,F2 = PPTC Resettable Fuse, smd, polyfuse, 1210L050YR Littelfuse
PCB 150719-2 in Elektor store [7]



deerde SIL-headers aangesloten worden op het externe circuit.
De board-to-board connector met 32 contacten van Harwin die we in ons T-board prototype hebben gebruikt om de print aan te sluiten op een breadboard moet gesplitst worden in twee 16-pens stekers (K2 en K3). Elk contact heeft een dikkere en dunnere pen. Het dikkere deel van de contacten moet aan de print worden gesoldeerd, zodat de dunnere pen in het breadboard komt (en daar de contactrails niet beschadigt of vervormt).

Test-software

Er zijn twee pythonprogramma's beschikbaar [5] waarmee u een functionele test kunt doorvoeren. Het ene programma dient om alle GPIO's als uitgang te testen (Check_all_GPIOs_as_output.py) en met het andere, u raadt het al, kunt u alle GPIO's als ingang testen (Check_all_GPIOs_as_input.py).
Bij het testen van de GPIO's als uitgang zijn slechts acht low-current LED's nodig met het python-programma. De uitgangen worden namelijk getest in groepen van acht.
Als voorschakelweerstand voor de LED's kan 1,8 kΩ worden gebruikt, maar deze waarde is niet zo kritisch. De weerstand voorkomt beschadiging van de LED als deze direct op de plus wordt aangesloten. Door de open drain-uitgang is de spanning over een (rode) LED plus weerstand ongeveer 2,6 V als de 5 V is geselecteerd als voeding voor de uitgangen (met JP3). Als de LED's naar verwachting oplichten, is de schakeling getest en klaar voor gebruik.

Let op de belasting

Nog een laatste opmerking: de TXS0108E is ontworpen voor belastingen met een hoge-impedantie. Hij heeft een interne serieweerstand van ongeveer 4k en kan geen grote stromen leveren. Als de belastingsimpedantie te laag is, oscilleert het uitgangssignaal. Houdt dit dus in gedachten wanneer u iets op de buffer aansluit. Meer info hierover vindt u in de application note [4] van het IC.

(180430)



IN DE STORE

→ Print 150719-1
www.elektor.nl/raspi-pcb-150719-1

→ Print 150719-2 (T-Board)
www.elektor.nl/raspi-pcb-150719-2

Weblinks

[1] Webpagina van Brian Dorey: www.briandorey.com/post/raspberry-pi-gpio-protection
[2] Voorstel van de auteur op de Labs-website: www.elektormagazine.com/labs/raspi-buffer-board
[3] Labsproject-website: www.elektormagazine.nl/labs/buffer-boards-for-raspberry-pi-23
[4] Technische documentatie over de TXS0108E: www.ti.com/product/TXS0108E/technicaldocuments
[5] Projectpagina op de magazinewebsite: www.elektormagazine.com/180430-01
[6] PCB voor Raspberry Pi Buffer Board: www.elektor.nl/raspi-buffer-150719-1
[7] PCB voor Raspberry Pi Buffer Board (T-Board): www.elektor.nl/raspi-buffer-150719-2



Project 2.0

correcties, updates en brieven van lezers

Experimentele Doppler-radar

Elektor juli/augustus 2018, p. 48 (160385)

De print die u bij Elektor koopt, is versie v1.2 of v1.3 (elektrisch identiek). Ten opzichte van de bij het artikel afgedrukte versie v1.1 bestaan de volgende verschillen:

R45 (47 Ω) is nieuw en staat daarom ook niet in de onderdelenlijst van het project. Deze weerstand helpt de schakelruis van IC11 te reduceren. R44 in de eerste versie is R42 in v1.2/v1.3, terwijl R44 in v1.2/1.3 nu R42 is (gevolg van een ongelukkige hernummering van onderdelen). R44 dient alleen voor testdoeleinden en moet niet worden gemonteerd. R42 moet alleen worden gemonteerd wanneer IC11 in plaats van IC4 wordt gebruikt (zie verderop).

Voor de volledigheid: alle passieve componenten zijn 0805 voor zover niet anders aangegeven. L1 is type SRR4828A-150M van Bourns of equivalent. Het volledige typenummer van IC4 luidt ADP2370ACPZ-3.3-R7, wat in de tekst van het artikel is afgekort tot ADP2370-3.3. De ADP2370 is de instelbare versie die in deze schakeling niet functioneert.

3,3-V-spanningsregelaar: zoals in het artikel al opgemerkt, bestaan er voor de 3,3-V-regelaar twee mogelijkheden. Slechts één van beide monteren en NIET allebei!

In de onderdelenlijst is tot onze spijt een foutje geslopen: IC4 moet zijn ADP2370ACPZ-3.3-R7, terwijl IC5 de ADP150AUJZ-3.0-R7 is.

Optie 1 (moeilijk te solderen, minder onderdelen)	**Optie2** (eenvoudiger te solderen, meer onderdelen)
IC4, R29	C19, C20, C21, D2, D3, IC11, R42, R43, R45
De onder optie 2 genoemde onderdelen NIET monteren	De onder optie 1 genoemde onderdelen NIET monteren



10-MHz-referentie

Elektor mei/juni 2018, p. 32 (160594)

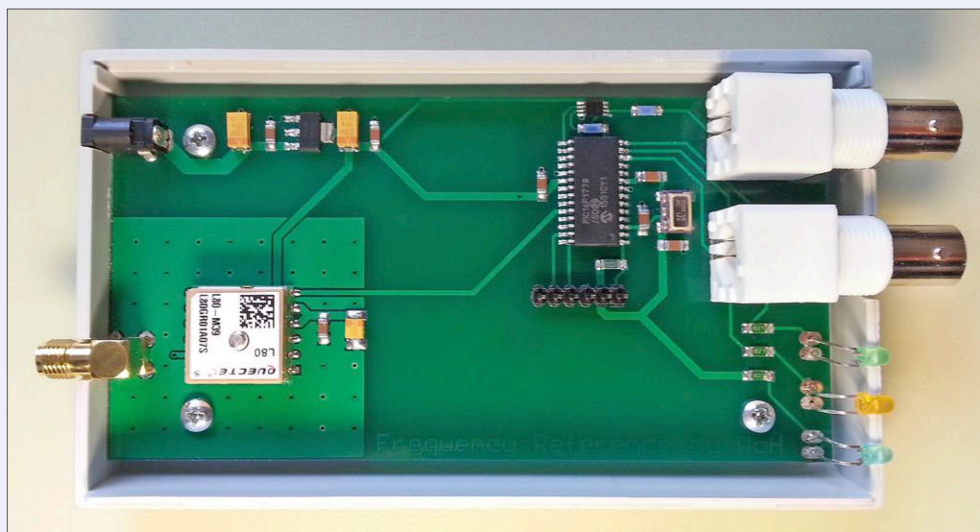
Uw artikel over een GPS-gestabiliseerde referentiefrequentie past precies in mijn straatje — het stabiliseren van een 10-GHz-LNB. Helaas heb ik voor die LND niet 10 MHz nodig maar 24 MHz. Daarom mijn vraag of in plaats van het 20-MHz-kristal ook een 48-MHz-VXCO gebruikt kan worden en de teller in dat geval niet tot 160.000.000 maar tot 384.000.000 zou tellen voordat vergelijking plaatsvindt? Is de hogere ingangsfrequentie voor de PIC16F een probleem of is die hogere waarde voor de teller een probleem? Welke wijzigingen in de firmware zouden nodig zijn?

Karl-Gerhard Ruthemann

De auteur antwoordt:

Geen van de PIC's die ik ken, kan een externe klokfrequentie van meer dan 20 MHz aan. In uw geval zou men de frequentie van de VCTCXO door vier kunnen delen en de processor op een klokfrequentie van 12 MHz laten draaien. Men moet dan de interne teller aanpassen en een buffer tussen VCTCXO en uitgang opnemen. Dat zou niet al te moeilijk mogen zijn.

Willem den Hollander





Special Quality (SQ) Buizen (Retrotronica)

Elektor september/oktober 2018, p. 104 (180299)

Als lezer van het eerste uur verheug ik me altijd weer over dit soort artikelen. En in dit verband moest ik denken aan een buis die ik in de jaren '70 als buisvervanger heb gekocht (zie foto). Wanneer u hiermee uw verzameling wilt uitbreiden, stuur ik hem graag naar u op.

Dieter Becker



De PL802E op de foto is een halfgeleider-vervanger (de "E" staat voor "Ersatz") voor de PL802-buis. Ik vermoed dat er een hoogspanning-transistor in zit. Hij heeft zodoende niets van doen met de in het artikel besproken "Special Quality"-buizen. Maar ik zou dit exemplaar toch graag willen toevoegen aan mijn verzameling van zeldzame buizen en oude apparaten, die ik bezoekers aan ons lab in Aken af en toe toon.

Jan Buiting



Special Quality (SQ) Buizen (Retrotronica)

Elektor september/oktober 2018, p. 104 (180299)

Ik heb een grote verzameling buizen en bouw daarmee ook veel buizenversterkers, voornamelijk in OTL-technologie (Output Transformerless). Ook heb ik veel computerbuizen uit rekenapparatuur van NCR (Computronic) uit de jaren '50. Daaronder de 5751 Command van RCA, de 5963 Sylvania Gold maar ook de E81CC van Siemens (alsmede E83CC en E88CC).

Mijn dank aan het Retrotronica-team voor de altijd weer interessante bijdragen!

Werner Frick

Dank u voor uw enthousiaste antwoord! Over dit artikel heb ik een heleboel feedback van Elektor-lezers mogen ontvangen. Oorspronkelijk wilde ik ook nog een paar alinea's aan de testprocedures wijden die door Valvo Duitsland voor de SQ-buizen zijn ontwikkeld, maar dan zou het artikel te lang zijn geworden. Zelf ben ik (ook) verzamelaar van OTL-versterkers (zij het beperkt tot typen met 800 Ω en 1200 Ω impedantie). Ik geloof dat het OTL-principe in de audio-buizenwereld veel te weinig aandacht krijgt. Voor een komende uitgave van Elektorlabs schrijf ik misschien een artikel over de Philips 1200-Ω-Circlotron-versterker.

Jan Buiting

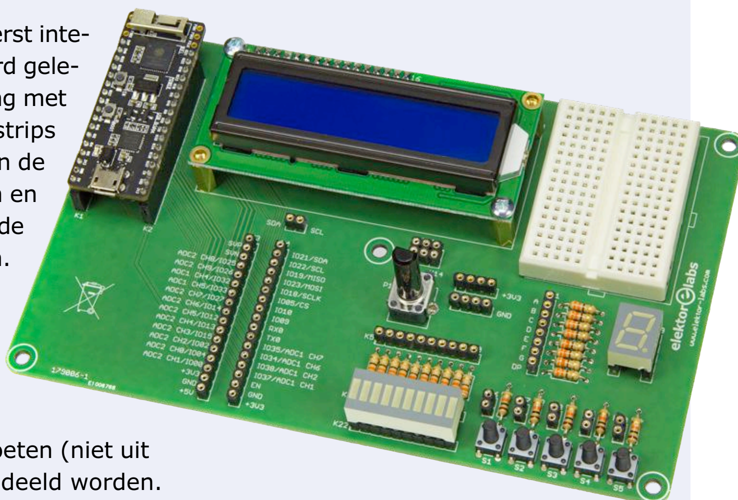


Het ESP32 Pico-Discovery-Board

Elektor september/oktober 2018, p. 32 (180341)

Ik heb in het jongste Elektorlabs-nummer het uiterst interessante artikel over het ESP32 Pico_Discovery-Board gelezen; in dit verband zou ik graag een praktische ervaring met u willen delen. Op het board zijn veel aansluitingen op busstrips naar buiten geboerd. Deze busstrips zijn niet echt goedkoop in de elektronica-shop. Ik kon daarom de verleiding niet weerstaan en heb ze besteld bij een verkoper op AliExpress — 10 stuks voor de prijs van één hier in de elektronicawinkel, geen verzendkosten. Die strips zien er precies hetzelfde uit, maar bij gebruik van normale draad van 0,6 mm blijkt dat die er na twee- of driemaal insteken/uittrekken gelijk weer uit valt — de veercontacten zijn kennelijk moe. Ik heb ze meteen weggegooid, dat geeft teveel ergernis.

Ik heb daarentegen wel goede ervaringen met 40-pens IC-voeten (niet uit het Verre Oosten); die kunnen met een scherp mes goed gedeeld worden.



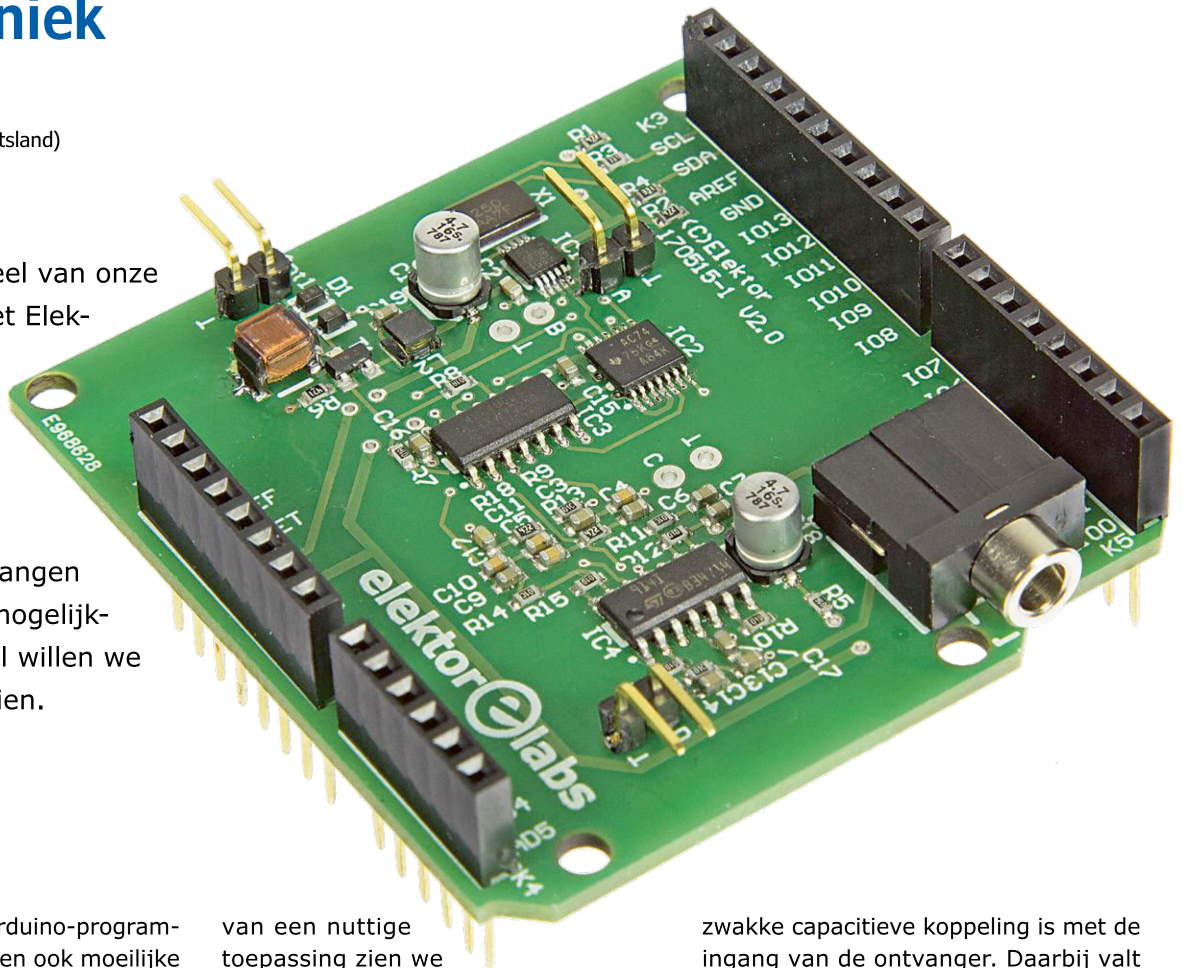
Ruedi Heimlicher

Elektor SDR-Shield 2.0 (3)

meettechniek

Burkhard Kainka (Duitsland)

Dit is het laatste deel van onze kleine serie over het Elektor SDR-Shield 2.0 [1]. Een automatisch afstembare ontvanger en twee onafhankelijk bestuurbare VFO-uitgangen openen ontelbare mogelijkheden. In dit artikel willen we er een paar laten zien.

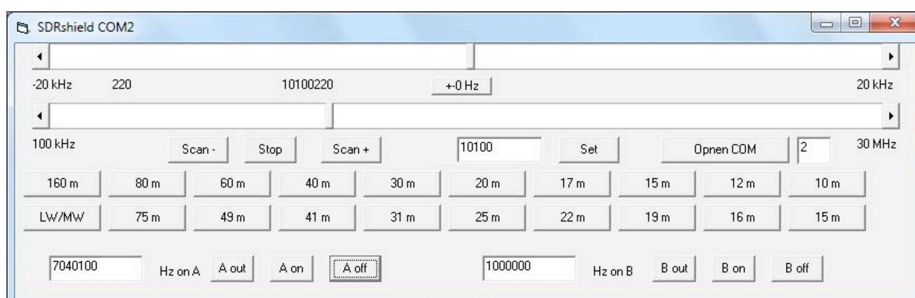


Door wat eenvoudige Arduino-programma's te schrijven, kunnen ook moeilijke vraagstukken worden opgelost. Het is niet altijd nodig om een duur meetapparaat aan te schaffen, we kunnen ook programmeren wat we nodig hebben.

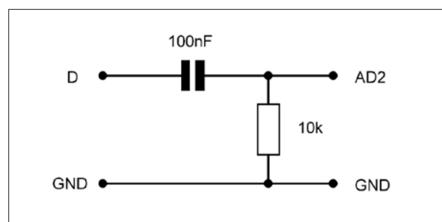
Hulpdraaggolf

Zodra één van de extra kanalen wordt ingeschakeld, verschijnt er een zwak signaal in de ontvanger. Een voorbeeld van een nuttige toepassing zien we al bij de ingestelde frequentie voor kanaal A. 7040,100 kHz is het midden van het slechts 200 Hz brede WSPR-bereik in de 40m-band (**figuur 1**). Bij de nauwkeurige afstemming kan de hulpdraaggolf helpen bij de oriëntatie. Zelfs zonder een kabel aan te sluiten, wordt de ingeschakelde draaggolf zichtbaar in de ontvanger omdat er altijd een zwakke capacitieve koppeling is met de ingang van de ontvanger. Daarbij valt op dat er geen volledige zijbandonderdrukking wordt bereikt. Het signaal verschijnt misschien bij +12 kHz en iets zwakker nog een keer bij –12 kHz. Dat komt omdat het niet alleen op de ingang inkoppelt, maar ook op andere plaatsen in de ontvanger, zoals bij de uitgang van de mengtrap.

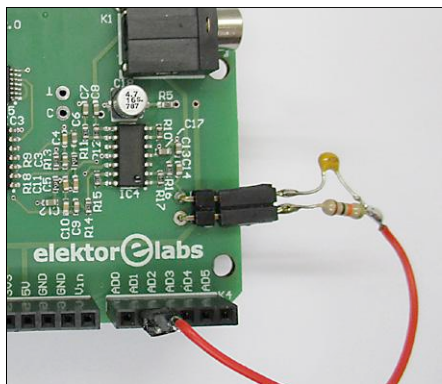
Ook harmonischen kunnen worden ontvangen. Omdat er symmetrische blokgolfsignalen worden opgewekt, zijn alle oneven harmonischen sterker. Een uitgangssignaal van 1 MHz is daardoor ook te zien bij 3 MHz, 5 MHz, 7 MHz enzovoort. Normaal gesproken zullen we alle ongemoduleerde draaggolven ontvangen met de instelling USB of LSB. Als er twee dicht bij elkaar liggende draaggolven zijn, dan is ook AM te gebruiken. We stellen bijvoorbeeld 20000 kHz in op kanaal A en 20001 kHz op kanaal B. Dan ontvangen we op 20 MHz een AM-signaal,



Figuur 1: Instelling van de VFO-frequentie en de extra uitgangen.



Figuur 2: Meten van het signaalniveau.



Figuur 3: Verbinding tussen de LF-uitgang en de A/D-ingang.

Listing 1: Meten van het bandgebruik (programma RFplotdB).

```
void uac(void)
{
  long uac =0;
  for (int i=0; i <= 200; i++){
    uac = uac+ analogRead(A2);
   }
  //if (uac > 400) uac = 400;
  u=20 * log10(uac);
  Serial.println(u);
}

void loop(void)
{
  for (int i=1; i <= 498; i++){
    int f= i * 50;
    si5351.set_freq((f+5)*400000, SI5351_PLL_FIXED, SI5351_CLK1);
    if (i== 1) delay (500);
    delay(10);
    uac();
  }
  while ( (digitalRead(A1) == 1));
}
```

dat gemoduleerd is met 1 kHz. Dit soort schijnbaar nutteloze spelletjes kunnen helpen om de achtergronden, mogelijkheden en grenzen van de hardware nog beter te leren kennen. In de praktijk kunnen we dan ook onverwachte problemen begrijpen. En we kunnen de harmonischen met hun bekende signaalsterktes gebruiken als referentiesignalen om de signaalsterkte-indicatie van ontvangers te kalibreren (zie onder).

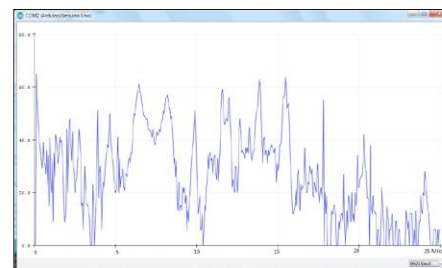
Metingen aan microcontrollersystemen

Een systeem met een ontvanger en generatoren kan in veel situaties goed van pas komen. We kunnen dat illustreren met een voorbeeld uit de dagelijkse praktijk in het elektronicalab. Een systeem met twee controllers had communicatiefouten op de seriële interface tussen beide controllers. Daarom werden om te beginnen de klokfrequenties nagemeten. Maar je kunt niet zomaar een frequentieteller aansluiten, omdat die de te onderzoeken oscillator zou beïnvloeden. Dus werden de signalen afgeluisterd via een extreem losse koppeling met de SDR: er werd gewoon een antennedraad dicht bij de oscillator gehouden. De keramische oscillator van 12 MHz konden we rechtstreeks ontvangen en die werkte binnen de gespecificeerde toleranties. De tweede controller had een 40-MHz-kristal en werd afgeluisterd bij 13,333 MHz door het mengen van harmonischen. Ook deze frequentie klopte heel nauwkeurig. Maar toch toonde de oscilloscoop aan, dat de baudrate van de snelle controller ongeveer 5 % te hoog was. Er moest dus sprake zijn van een programmeerfout. Een tweede bewijs daarvoor werd gevonden door de 12-MHz-oscillator van de langzame controller een extern kloksignaal op te dringen. Daar werd uitgang A van het SDR-shield voor gebruikt. Het bleek dat we de frequentie tot minstens 12,3 MHz moesten opvoeren voordat de seriële communicatie foutloos werkte. Zo heeft het SDR-shield geholpen om dit probleem te analyseren.

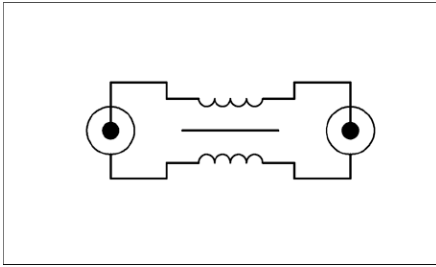
Weergave van het bandgebruik

De omstandigheden voor de propagatie van radiogolven op korte golf veranderen van uur tot uur. Het is daarom interessant om te weten welke banden veel gebruikt worden op welke tijd en dus goede propagatiecondities voor radiogolven bieden. Daarvoor hebben we een automatisch afstembare ontvanger en een meting van de signaalamplitude nodig. In dit geval is geen volledige decodering van het IQ-signaal nodig, maar hoeven we slechts de signaalspanning aan een van de uitgangen te meten. Dat kan de Arduino alleen aan als het om niet al te zwakke signalen gaat. Het LF-signaal op aansluiting D wordt op een gemiddeld GND-potentiaal gebracht en rechtstreeks gemeten door de A/D-converter op kanaal ADC2 (**figuren 2** en **3**). Omdat de A/D-converter alleen positieve spanningen meet, wordt er in feite gelijkgericht. De gemiddelde waarde uit 200 aparte metingen is een betrouwbare maat voor de signaalspanning.

Het programma *RFplotdB* (**listing 1**, alle programma's kunnen worden gedownload van [2]) bevat een ontvangstlus voor 0...25 MHz. De signaalspanningen op de uitgang van het shield worden gemiddeld, gelogaritmiseerd en weergegeven in dB. Het resultaat kon worden weergegeven met de seriële plotter in de Arduino-IDE. Omdat het venster plaats



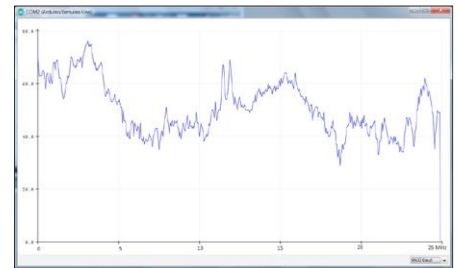
Figuur 4: Het spectrum bij een vrijhangende dipool.



Figuur 5: Principeschema van het mantelstroomfilter.



Figuur 6: Coaxkabel op een ferrietkern.



Figuur 7: Signalen van een binnenantenne.

biedt voor 500 meetwaarden, kunnen we een frequentiebereik tot 25 MHz weergeven met een oplossend vermogen van 50 kHz. Dat komt ook ongeveer overeen met de bandbreedte van de ontvanger zonder achtergeschakelde SDR-software. In **figuur 4** zien we het bandgebruik in de middag voor een vrij opgespannen dipool van twee keer 7,5 m lang. We hebben daarbij een mantelstroomfilter gebruikt om storingen uit het gebouw weg te houden van de ingang van de ontvanger. Zo'n filter bestaat uit een ferriet-ringkern met daarop meerdere windingen van dunne coaxkabel (**figuren 5** en **6**). Deze spoel houdt de op de buitenste geleider ingevangen storingen uit het gebouw weg van de antenne. We kunnen zien dat dit goed werkt aan het dynamische bereik van 60 dB in de meetresultaten.

Ter vergelijking ziet u in **figuur 7** een meting aan een binnenantenne. Dit was een lusantenne die bestond uit in totaal 30 m draad. Hier en daar zijn wel signalen met een vergelijkbaar niveau te herkennen, maar nergens in het spectrum komt het ruisniveau onder 30 dB. Het meetresultaat komt bij veel frequenties overeen met de werkelijke ontvangstresultaten, waarbij overal een te hoge ruisvloer de ontvangst verstoorde.

Listing 2: Vierpoolmeting tot 25 MHz (RFplotU)

```
void uac(void)
{
  long uac =0;
  for (int i=0; i <= 200; i++){
    uac = uac + analogRead(A2);
  }
  u = uac/50;
  Serial.println (u);
}


void loop(void)
{
  Serial.println (0);
  for (int i=1; i <= 500; i++){
    int f= i * 50;
    //Serial.print (f); Serial.print (“  “);
    si5351.set_freq(f*100000, SI5351_PLL_FIXED, SI5351_CLK0);
    delay(10);
    si5351.set_freq((f+5)*400000, SI5351_PLL_FIXED, SI5351_CLK1);
    if (i== 1) delay (500);
    delay(10);
    lcd.setCursor(0, 0);
    lcd.print (f);
    lcd.print (“  “);
    delay(20);
    uac();
  }
  while ( (digitalRead(A1) == 1));
}
```

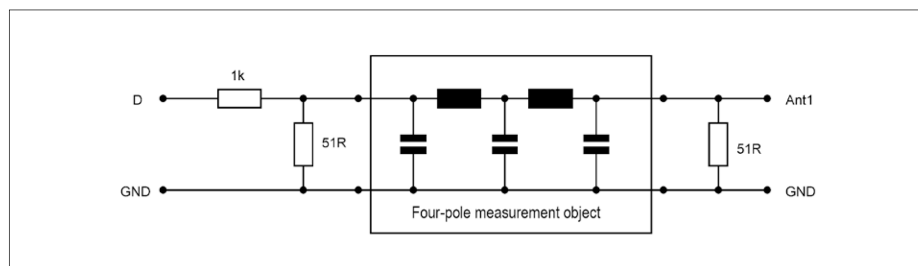
Vierpoolmetingen

Om passieve vierpolen of tweepolen te onderzoeken in een brede frequentieband, hebben we een afstembare oscillator met een meelopende ontvanger nodig. Dat is met het SDR-shield gemakkelijk te realiseren. We breiden de al beschikbare ontvangstlus voor 0 tot 25 MHz uit met een VFO-uitgang, die meeloopt op precies 5 kHz naast de ontvangstfrequentie (**listing 2**). De ontvanger wordt daarbij gebruikt als een directe mengtrap, waarbij een 5-kHz-signaal aan de uitgang de amplitude van het ontvangen signaal aangeeft.

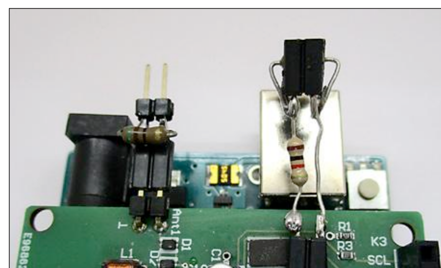
Voor eenvoudige vierpoolmetingen hebben we alleen een spanningsdeler nodig, die de signaalamplitude terugbrengt tot een voor de ontvanger toegestane ingangsspanning van ongeveer 200 mV. We gebruiken een spanningsdeler van 1 kΩ en 51 Ω (**figuren 8** en **9**), zodat de generator een impedantie van ongeveer 50 Ω krijgt. Afhankelijk van het meetobject kan het nodig zijn om ook de ingang van de ontvanger af te sluiten met een passende belastingsweerstand. In het voorbeeld meten we aan een laagdoorlaatfilter voor een kortegolfzender met een impedantie van 50 Ω.

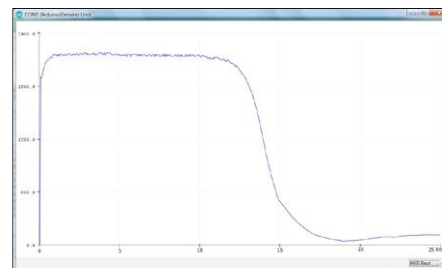
Het meetresultaat in **figuur 10** laat zien dat afsnijfrequentie bij ongeveer 13 MHz ligt. De demping is duidelijk meer dan 10 maal (20 dB) bij ongeveer 18 MHz.

Dat de uitgangsspanning vanaf 20 MHz weer licht toeneemt, kan komen door de zelfinductie en spanningsval in de massalijnen, die in dit geval bestonden uit niet-afgeschermde kabels. Bij veel metingen blijkt, dat in dit frequentiebereik kabellengtes van enkele centimeters al kunnen leiden tot duidelijk zichtbare meetfouten, tenzij overal coaxkabels met de juiste golfweerstand en afgeschermde behuizingen worden gebruikt.
Als we het LF-signaal aan de uitgang van de ontvanger onderzoeken met een oscilloscoop, vinden we een driehoekgolf. Dat klopt: bij het mengen van twee blokgolven ontstaat inderdaad altijd een driehoek, omdat de beide rechthoeken met een constante faseverandering over elkaar schuiven. We kunnen het ook zó interpreteren, dat het uitgangssignaal boventonen bevat, die in de ontvanger met elkaar worden gemengd. Als het ingangssignaal door een voldoende krachtig laagdoorlaatfilter wordt gevoerd, verschijnt er een sinus aan de LF-uitgang. Het gebruik van blokgolven leidt tot meetfouten door de boventonen, maar in de meeste gevallen kan de invloed daarvan worden verwaarloosd.



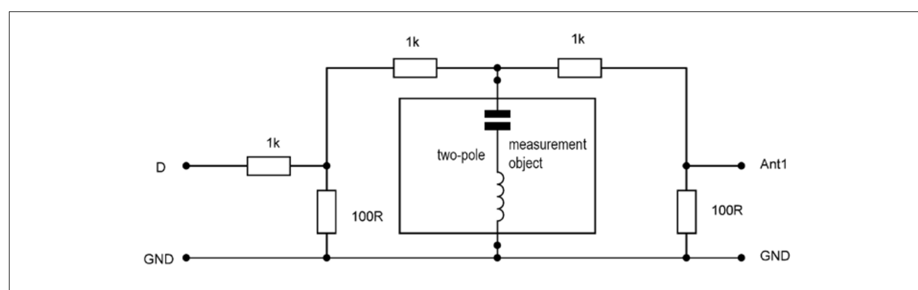
Figuur 8: Vierpoolmeting aan een laagdoorlaatfilter.


Figuur 9: Connectoren en weerstanden voor de vierpoolmeting.

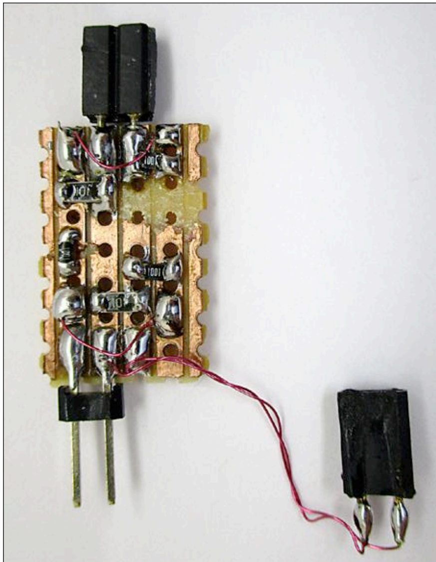

Figuur 10: Frequentiekarakteristiek van een laagdoorlaatfilter.

Kalibratie van de S-meter

De S-meter van een kortegolfontvanger moet bij 50 µV op de antenne-ingang S9 aangeven. Elke S-stap komt overeen met 6 dB. S8 = 25 µV, S7 = 12,5 µV, S6 = 6 µV, S5 = 3 µV enzovoort. Maar de nauwkeurigheid van de meeste S-meters is niet zo groot. We hebben dus een betrouwbaar HF-signaal met een bekend niveau nodig. Bij een directe verbinding van de verzwakker aan uitgang A met de antenne-ingang met een parallelle 50Ω-weerstand (verzwakker zoals in figuur 8) hebben we een signaal met een gedefinieerd niveau. Zo kunnen we een S-meter kalibreren of de gevoeligheid van de SDR afregelen door de instelling van de geluidskaart.
Het blokgolfsignaal op uitgang A heeft altijd een spanning van 3,3 Vtt, dus een amplitude van 1,65 V. De verzwakker met 1 kΩ en 25 Ω heeft een deelfactor van 41, dus krijgen we 41,2 mV. Dat is nog steeds veel meer dan een gebruikelijk antennesignaal en komt overeen met ongeveer S9 + 60 dB.
We kunnen dat oplossen door te kijken naar de ontvangst van de veel zwakkere harmonischen. Volgens Fourier bevat een blokgolfsignaal met een amplitude A de basisfrequentie f0 en alle oneven boventonen (f0*k) met een amplitude 4 A/(π * k). De basisfrequentie heeft dus in dit geval een sinus-amplitude van 52,5 mV, wat overeenkomt met een effectieve spanning van 37,1 mV. Maar als we de duizendste harmonische ontvangen, is het niveau nog maar 37,1 µV. Om precies te zijn, moeten we het signaal zoeken op f0 * 999 of op f0 * 1001, want alle even harmonischen zijn bij een blokgolfsignaal nul.
In concreto: we stellen de frequentie voor uitgang A in op 10 kHz in en we vinden bij 10010 kHz een signaal van 37 µV. Of we zoeken een harmonische bij 7 MHz (om precies te zijn bij 7430 kHz) en komen uit op 50 µV, dus precies S9. Als we de VFO van de ontvanger instellen op 7000 kHz, verschijnen er veel sterkere signalen bij −10 kHz en bij +10 kHz. Die signalen zijn afkomstig van de grondtoon van 10 kHz en zijn door de mengtrap heen gekomen. Maar zodra de ontvangstfrequentie met 1 kHz wordt verschoven naar 7001 kHz, vinden we het gezochte signaal bij +9 kHz (harmonische op 7010 kHz) en bij −11 kHz (harmonische op 6990 kHz).
Op dezelfde manier vinden we verder harmonischen op een afstand van telkens 20 kHz. Al die signalen hebben een spanning van 50 µV en zouden dus als S9 moeten worden weergegeven. Hiermee kan de ingangsgevoeligheid van de SDR worden ingesteld. Een zinnig niveau voor het S9-signaal is −10 dB. Een signaal met een S1-niveau zou dan 8 * 6 dB = 48 dB lager liggen bij ongeveer −60 dB. Dat is ook ongeveer het niveau van de achtergrondruis in de ontvanger zonder aangesloten antenne.
Het signaal met veel harmonischen past heel goed bij de amateurradiobanden. Als



Figuur 11: Tweepoolmeting.



Figuur 12: Inductiearme opbouw met SMD-componenten.

we op 7 MHz S9 krijgen, is het bij 14 MHz nog S8 en bij 28 MHz S7. Metingen met het SDR-shield bevestigen deze niveaus, wat opnieuw bewijst dat de gevoeligheid van de ontvanger over het hele bereik tot 30 MHz constant is.

Tweepoolmetingen

We kunnen ook de frequentiekarakteristiek van een impedantie meten. De eenvoudige schakeling in **figuur 11** werkt met een hoogohmige spanningsdeler die bestaat uit een weerstand van 1 k en het meetobject. Via een tweede spanningsdeler gaat het signaal naar de ingang van de ontvanger. De schakeling is met SMD-weerstanden zo inductiearm mogelijk opgebouwd (**figuur 12**). Met deze eenvoudige schakeling kunnen we geen onderscheid maken tussen de reële weerstand en de blindweerstand. We meten de impedantie in ohm. Een gemeten weerstand van 100 Ω zou dus reëel kunnen zijn (een ohmse weerstand) of bijvoorbeeld capacitieve reactantie.
De gemeten spanning wordt omgerekend naar de weerstand van het te meten object en serieel uitgevoerd (**listing 3**). Daarnaast wordt continu de momentele frequentie weergegeven op het LCD. Met druktoets S2 op het LCD-shield wordt een nieuwe cyclus gestart. Het hele impedantieverloop wordt opgeslagen in het array d[], zodat we ook zonder verbinding met de PC bijvoorbeeld rechtstreeks aan het voetpunt van een antenne kunnen meten. Druk op S1 om de opgeslagen data weer te geven.
Voor een autonome meting moeten we een externe voeding aansluiten. Verder moeten we de resetlijn van de Arduino blokkeren met een brug naar +5 V. Dat is nodig omdat op het moment dat we de kaart weer aansluiten op de PC en de seriële plotter starten, deze probeert het board te resetten, wat de data in het RAM-geheugen zou wissen. Door de reset-functie te blokkeren blijft de data behouden en kunnen we die met S1 naar de plotter sturen.
In **figuur 13** zien we het meetresultaat: het impedantieverloop van een pi-filter dat bestaat uit twee draaicondensatoren en een spoel. We zien hier een serieresonantie met de kleinste impedantie bij 7 MHz en een parallelresonantie met een maximumimpedantie bij 11 MHz.
In **figuur 14** zien we het impedantieverloop van een provisorische binnenantenne. Hij bestaat uit een ongeveer 30 m lange en maar 0,2 mm dikke gelakte koperdraad die in een zo groot mogelijke lus aan de muur bevestigd is en in totaal twee kamers omsluit. We zien een duidelijke resonantie iets onder 10 MHz, zoals te verwachten is voor een quad-antenne met een lengte van een hele golflengte. Bij het resonantiepunt bedraagt de impedantie ongeveer 100 Ω. Er is een tweede resonantie te zien bij ongeveer 18 MHz. Met een goede impedantie-aanpassing met een pi-filter is deze antenne goed te gebruiken als amateur-zendantenne, maar voor een ontvangstantenne produ-

Listing 3. Weergeven van een tweepoolmeting (fragmenten uit RFplotR).

```
void uac(void)
{
  long uac =0;
  for (int i=0; i <= 200; i++){
    uac = uac+ analogRead(A2);
  }
  if (uac>10000) uac=10000;
  r = uac*600/(14000-uac);
  //   Serial.print (uac);
  //   Serial.print (“  “);
  Serial.println (r);
}

void loop(void)
{
  for (int i=1; i <= 500; i++){
    int f= i * 50;
    // Serial.print (f); Serial.print (“  “);
    si5351.set_freq(f*100000, SI5351_PLL_FIXED, SI5351_CLK0);
    delay(10);
    si5351.set_freq((f+5)*400000, SI5351_PLL_FIXED, SI5351_CLK1);
    if (i== 1) delay (500);
    delay(10);
    lcd.setCursor(0, 0);
    lcd.print (f);
    lcd.print (“  “);
    delay (19);
    uac();
    d[i] = r;
  }
  while ( (digitalRead(A0) == 1));
  for (int i=1; i <= 500; i++){
    Serial.println (d[i]);
   }
  while ( (digitalRead(A1) == 1));
}
```

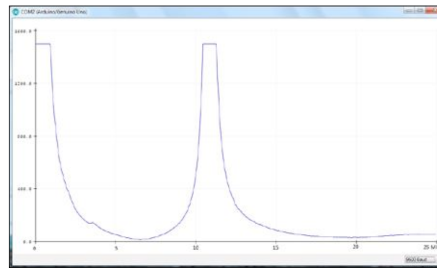
ceert hij helaas te veel achtergrondruis (vergelijk met figuur 7).

Staande golf-meetbrug

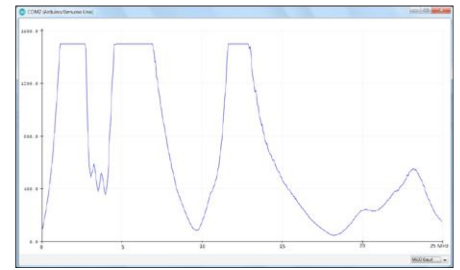
Een coaxkabel tussen antenne en ontvanger of zender heeft meestal een golfweerstand van 50 Ω. Als we dan aan het einde van de kabel een ohmse weerstand van 50 Ω aansluiten, zijn er geen reflecties: het hele in de kabel ingevoerde vermogen loopt dus maar in één richting. Maar als het einde van de kabel open wordt gelaten of wordt kortgesloten, ontstaat er een totale reflectie. De teruglopende golf wordt opgeteld bij de heenlopende golf, zodat er staande golven ontstaan. De kabel gedraagt zich dan als een soort resonantiekring, waarin de energie heen-en-weer wordt weerkaatst en volledig in warmte wordt omgezet door de ohmse kabelverliezen.

Een ideale dipoolantenne heeft bij zijn eigenfrequentie een impedantie van ca. 50 Ω. Dat levert een optimale aanpassing op en minimale verliezen. Bij een minder ideale antenne is er een afwijking van de gewenste weerstand en meestal ook nog een capacitieve of inductieve reactantie. Daardoor ontstaat er een gedeeltelijke reflectie met staande golven en grotere kabelverliezen. Dat is voor een ontvanger geen probleem, omdat de gevoeligheid bijna altijd groter is dan nodig. Maar veel zenders hebben wel een correcte impedantie nodig. Daarom wordt de staande golf-verhouding (SWR = *Standing Wave Ratio*) gemeten. Die geeft informatie over de afwijking van de golfweerstand van 50 Ω.

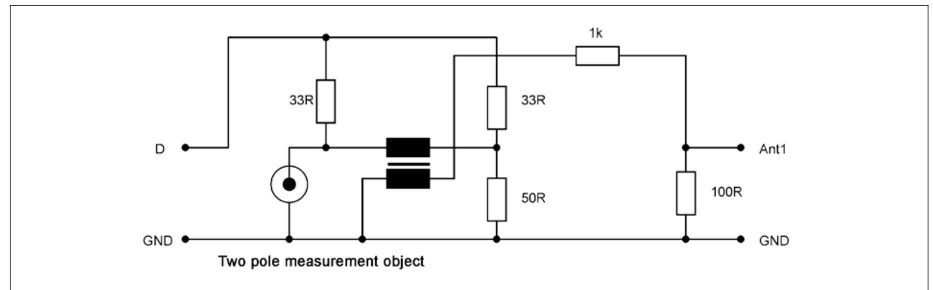
Deze meting kan met een eenvoudige meetbrug worden uitgevoerd (**figuren 15** en **16**). Als het te meten object een impedantie van precies 50 Ω heeft, is er geen brugspanning. Bij elke afwijking van de reële impedantie van 50 Ω ontstaat er een spanningsverschil tussen de takken van de brug. Dit wordt overgedragen via de kleine breedband-trafo in de brug. Bij een kortsluiting of een open aansluiting moet dezelfde hoge spanning worden gemeten als bij een oneindig grote SWR. De inwendige weerstand van de generator veroorzaakt wel een onbalans, die met kleine weerstanden in het bovenste deel van de spanningsdeler gedeeltelijk kan worden gecompenseerd. De gebruikte ringkerntransformator was trouwens toevallig gevonden in de knutselkist en oorspronkelijk niet bestemd voor dit frequentiebereik, maar hij bleek toch erg goed te werken.



Figuur 13: Impedantiekarakteristiek met serie- en parallelresonantie.

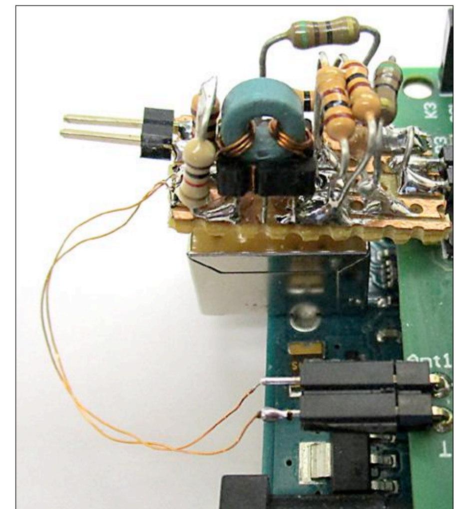


Figuur 14: Impedantiekarakteristiek een lusantenne.



Figuur 15: SWR-meetbrug.

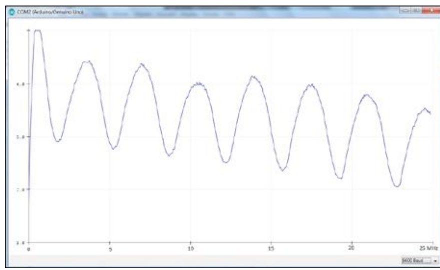
Een meting aan een 50-Ω-weerstand zou dus een spanning van nul volt moeten opleveren, wat overeenkomt met een SWR van 1,0. Met 100 Ω of 25 Ω zouden we een SWR van 2 moeten vinden. Metingen aan reële weerstanden (met korte aansluitdraden!) laten zien hoe nauwkeurig de meetbrug is. Er kunnen kleine afwijkingen ontstaan door de gebruikte transformator en door kabellengtes in de brugschakeling. Maar daar kunnen we wel mee leven, omdat het meestal alleen interessant is om te weten bij welke frequentie de SWR wijst op resonantie van de antenne, en hoe breedbandig de antenne is. In principe zijn deze vragen ook te beantwoorden met een eenvoudige impedantiemeting,



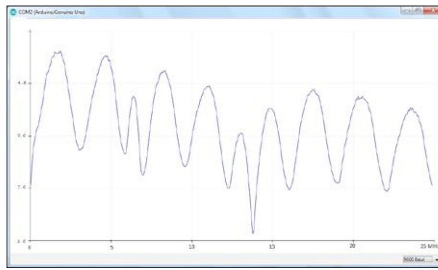
Figuur 16: Meetbrug met ringkerntransformator.

Listing 4: Weergeven van de SWR (programma RFplotSWR).

```
void uac(void)
{
  long uac =0;
  for (int i=0; i <= 200; i++){
    uac = uac+ analogRead(A2);
  }
  snr = uac/50;
  if (snr > 400) snr = 400;
  Serial.println (snr);
}
```

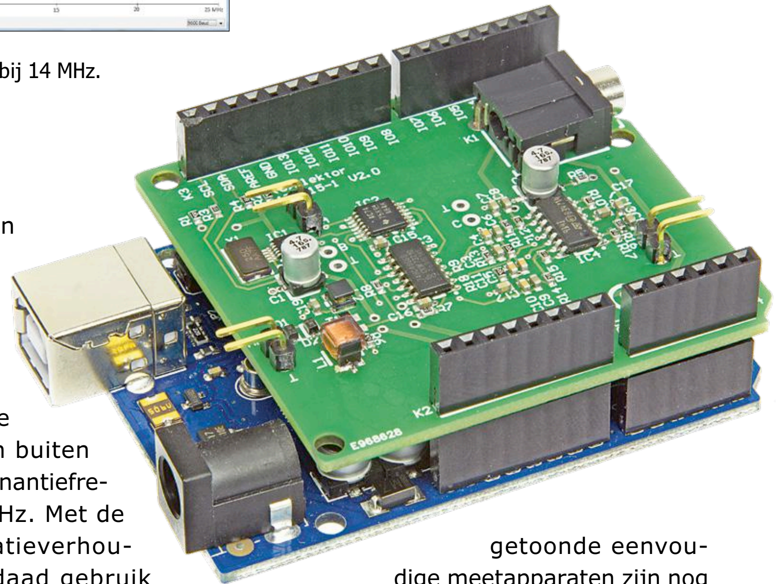


Figuur 17: SWR-meting aan een open antennekabel.



Figuur 18: Resonantie bij 14 MHz.

maar de SWR-meter herkent de ongewenste reactanties beter. De software (**listing 4**) is nauwelijks verschillend van die voor de eenvoudige meting van de frequentiekarakteristiek.
In **figuur 17** zien we het meetresultaat voor een open antennekabel. Op de punten met een kleine SWR hebben we weliswaar met een reële weerstand te maken, maar de kabel raakt in resonantie bij de oneven veelvouden van een kwart golflengte. Alle ingevoerde energie wordt in de kabel in warmte omgezet. Dat bij hogere frequenties schijnbaar betere SWR's tot ongeveer 2,0 worden bereikt, komt alleen door de grotere demping van de kabel bij hoge frequenties. Daar moeten we op letten om een aangesloten antenne goed te kunnen beoordelen.
In **figuur 18** zien we een meting aan een dipoolantenne van 2 x 7,5 m samen met zijn antennekabel. Aan het voetpunt van de antenne is een balun met een instelbare transformatieverhouding (**figuren 19** en **20**) gebruikt om de antenne te kunnen gebruiken buiten zijn eigenlijke resonantiefrequentie van 10 MHz. Met de juiste transformatieverhouding bleek inderdaad gebruik bij 14 MHz met een SWR van 1,2 mogelijk. En ook bij 7 MHz werd nog een SWR van ongeveer 2 bereikt. Zo zouden we op de 40-m-band en de 20-m-band dezelfde antenne kunnen gebruiken voor amateurradio-uitzendingen.
Ontvangers zijn trouwens veel minder kritisch, zodat ook een slecht aangepaste antenne nog steeds als ontvangstantenne kan worden gebruikt. Maar vaak blijkt, dat er alleen in het resonantiepunt een voldoende kleine antenne-impedantie is, waarbij ook het mantelstroomfilter goed werkt. Daarom worden in het resonantiepunt de beste storingsafstanden gehaald.
Veel van de hier getoonde eenvoudige meetapparaten zijn nog vatbaar voor verbetering, vooral wat betreft de opbouw en de connectoren. Maar ze tonen wel aan dat we al met weinig moeite bruikbare meetresultaten kunnen krijgen. Zo wordt het SDR-shield een veelzijdig hulpmiddel in het HF-lab. 

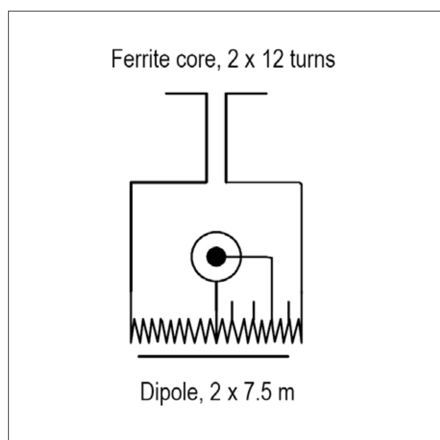


180284-B-04

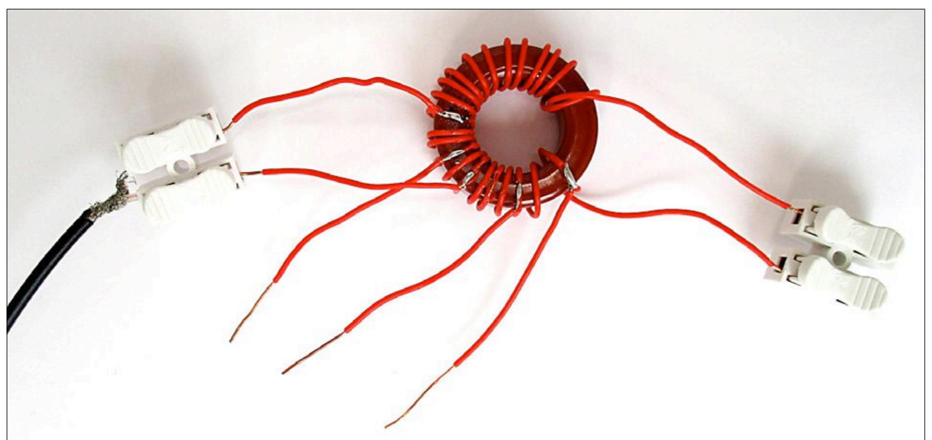
Weblinks

[1] Elektor SDR-Shield 2.0 (1): www.elektormagazine.nl/160577

[2] Download van alle programma's: www.elektormagazine.nl/180284-B-04



Figuur 19: Symmetrisch maken en aanpassen van de impedantie.



Figuur 20: Aanpassing en symmetrisch maken met een ijzerpoederkern-transformator.



Opamps in de praktijk

deel 1: principes

Burkhard Kainka (Duitsland)

Operationele versterkers vind je overal in de elektronica. Ze hebben uiterst uiteenlopende taken in de meest verschillende soorten schakelingen. Je hebt ze in allerlei typen en families, allemaal met hun eigen voor- en nadelen. Een nadere blik op de details is daarom zeker de moeite waard.

Operationele versterkers (opamps) zijn geïntegreerde schakelingen met bipolaire transistoren, JFET's of CMOS-transistoren. De naam 'operationele versterker' is afkomstig van hun oorspronkelijke toepassing als rekenversterker: met behulp van deze componenten konden analoge computers berekeningen uitvoeren zoals optellen, vermenigvuldigen enzovoort.

Opamp-basisschakelingen

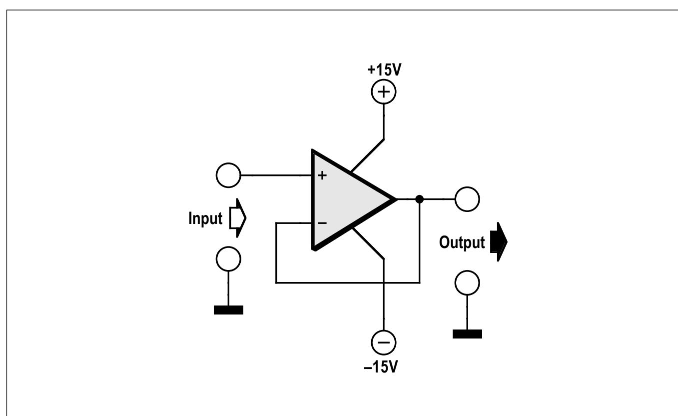
Een opamp versterkt het spanningsverschil tussen zijn beide ingangen. Een theoretische 'ideale' opamp heeft een oneindig grote spanningsversterking. In de praktijk zijn waarden tot zo'n 100.000 haalbaar. Meestal wordt een vorm van tegenkoppeling toegepast die het gedrag van de schakeling bepaalt. In **figuur 1** ziet u een volledig tegengekoppelde opamp. De spanningsversterking is precies 1, maar de stroomversterking is aanmerkelijk groter. De ingang is hoogohmig, de uitgang kan laagohmig worden belast. Deze schakeling wordt daarom meestal ingezet als buffer.

Spanningsvolger

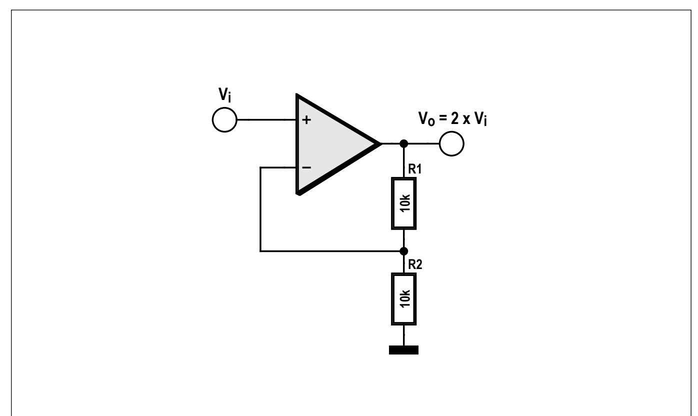
Men kan zich de werking als een regelkring voorstellen. De opamp vergelijkt continu de gewenste spanning op de niet-inverterende (+) ingang met de werkelijke spanning aan de uitgang en op de inverterende (–) ingang. De opamp probeert elk verschil tussen beide weg te werken. In de praktijk blijft er meestal een gering en constant verschil bestaan: de offset-spanning die in de orde-grootte van 1 mV ligt. Bij sommige typen, zoals de LM741, kan de offset op 0 worden afgeregeld. Vroeger was het gebruikelijk om opamps dubbelzijdig te voeden, meestal met +15 V en –15 V. Het toegestane spanningsbereik voor in- en uitgang liep dan minstens van –10 V tot +10 V. Deze conventie stamt uit de tijd van analoge computers, tegenwoordig wordt er meestal met lagere spanningen gewerkt.

Niet-inverterende versterker

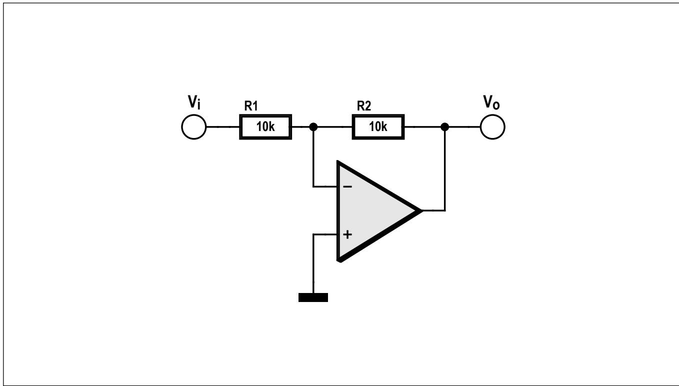
Een opamp is uitstekend geschikt om een bepaalde ingangsspanning met een exacte factor te versterken. Dat doen we door een spanningsdeler op te nemen in de tegenkoppeling, zie **figuur 2**. De uitgangsspanning stelt zich weer automatisch zodanig in dat de spanningen op de inverterende ingang (–) en op de niet-inverterende ingang (+) nagenoeg gelijk zijn. Elke kleine afwijking leidt namelijk tot een corresponderende verandering van de uitgangsspanning die door de tegenkoppeling wordt gecompenseerd. Door de hoge versterking verandert daarbij het verschil tussen de ingangsspanningen (de offset) praktisch niet.



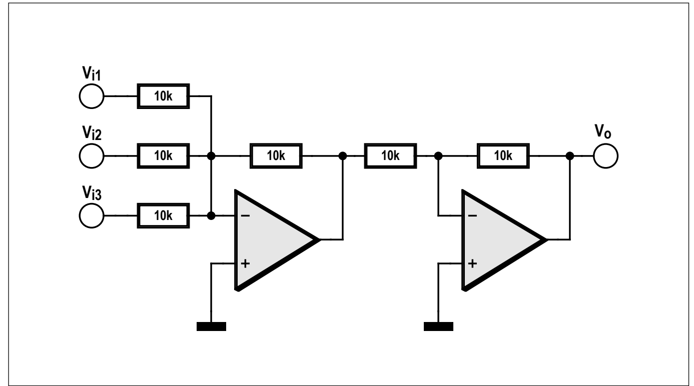
Figuur 1. Opamp als bufferversterker.



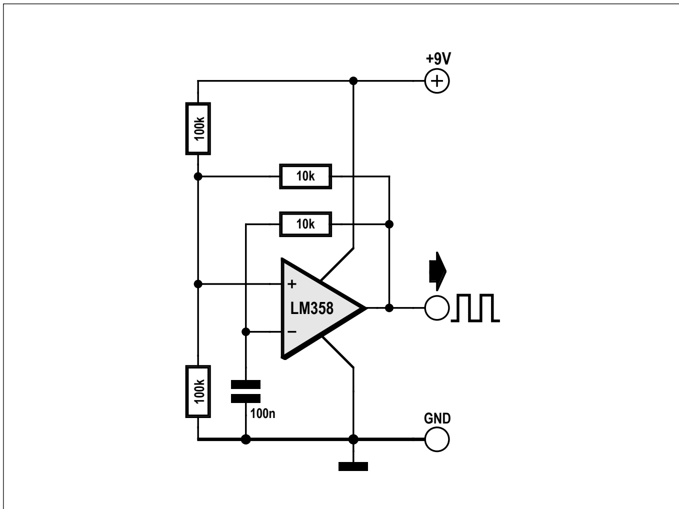
Figuur 2. Versterkingsfactor twee.



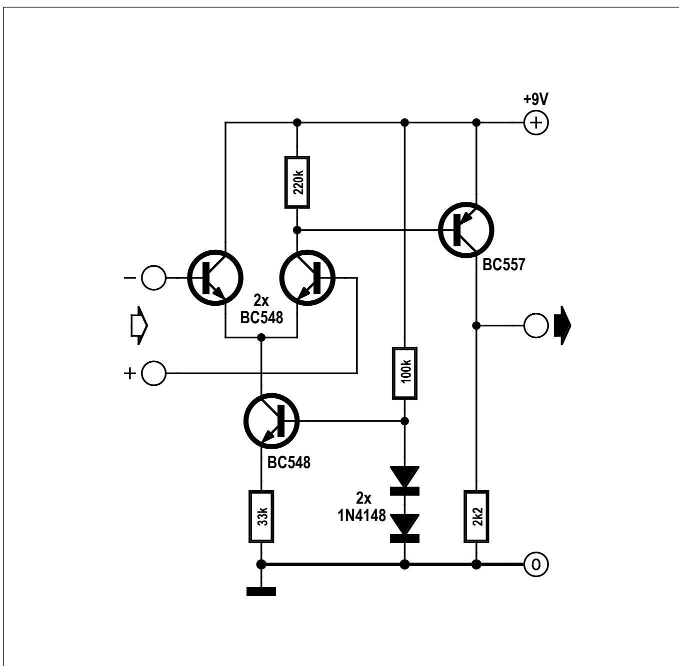
Figuur 3. Een inverterende versterker.



Figuur 4. Optellen van drie spanningen.



Figuur 5. Een blokgolfgenerator.



Figuur 6. Een opamp met discrete componenten.

Bij het ontwerpen van een schakeling gaan we vooralsnog uit van een ideale opamp, zonder offset-fout. Deze ideale opamp heeft bovendien een oneindig hoge ingangsweerstand, een uitgangsweerstand van nul en een oneindige bandbreedte. Bij de meeste toepassingen gedraagt een standaard-opamp zich vrijwel als een ideale opamp.

In principe kunnen we met de juiste keuze van weerstanden elke gewenste versterkingsfactor instellen. De versterkingsfactor is gelijk aan de deelfactor van de spanningsdeler (in figuur 2 dus een factor 2). Met een opamp kan een exacte versterking worden bereikt, zoals voor analoge computers vereist werd. Een schakeling voert dus een berekening uit, in dit geval een vermenigvuldiging met een constante factor V:

$$V = (R1 + R2) / R2$$

Bij heel hoge versterkingsfactoren moeten we er echter wel rekening mee houden dat de offsetfout van de opamp wordt meeversterkt. Enkele typen (bijvoorbeeld de oer-opamp 741) hebben speciale offset-null-aansluitingen. Andere typen zoals de OP07 worden in de fabriek afgeregeld op een offsetfout van hoogstens een paar microvolt.

Inverterende versterker

Een opamp kan worden gebruikt om een ingangsspanning exact te inverteren, zie **figuur 3**. De niet-inverterende ingang van de opamp ligt aan massa. De spanning op de inverterende ingang stelt zich eveneens in op nul volt. Met twee gelijke weerstanden krijgen we bij een ingangsspanning van +1 V een uitgangsspanning van –1 V, zodat de spanning op de inverterende ingang precies nul is.

Door andere weerstanden te kiezen, kan elke gewenste versterkingsfactor worden ingesteld. In het algemeen geldt:

$$V = -R2/R1$$

Opteller

Vaak is het nodig om meerdere ingangsspanningen bij elkaar op te tellen. Dat kan met een inverterende versterker met meerdere ingangsweerstanden (**figuur 4**). Met een nageschakelde tweede inverter erachter kunnen we de invertering van de eerste ongedaan maken.

Generator

Vinden we bij pure versterkerschakelingen altijd alleen maar een tegenkoppeling, bij generatorschakelingen is een meekoppeling (een positieve terugkoppeling) nodig. **Figuur 5** toont een blokgolfgenerator met een meekoppeling op de niet-inverterende ingang. Tegelijk is er een tegenkoppeling bestaande uit een RC-kring, die bepalend is voor de uitgangsfrequentie.

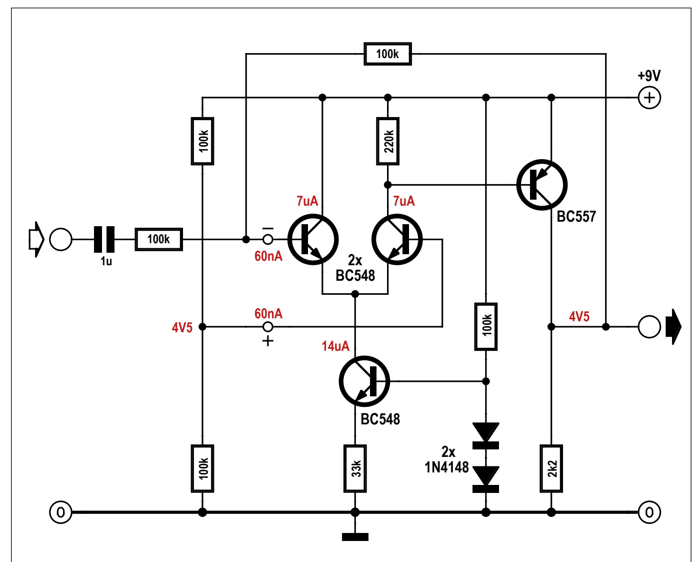
Opamps van binnen

In het inwendige van elke opamp vinden we een verschilversterker en een uitgangstrap. Je zou een vergelijkbare schakeling ook kunnen opbouwen met losse transistoren (**figuur 6**). Dat is een prima manier om de typische eigenschappen en de sterke en zwakke kanten van een opamp beter te begrijpen. Meting van de interne en externe spanningen in onze zelfgebouwde opamp in een reële toepassing met tegenkoppeling levert de waarde op die in **figuur 7** zijn ingetekend. De stroombron voor de beide ingangstransistoren levert een stroom van 14 µA, ongeveer gelijk verdeeld over beide ingangstransistoren. De rust-ingangsstroom bedraagt 60 nA (ter vergelijking: bij de 741 is dat 10 nA). De offsetspanning is ongeveer 5 mV (741: 1 mV). Tenslotte bepalen we nog de open-lus-versterking. Daartoe zetten we een (wissel)spanning van 2 Vtt op de ingang. Aan de uitgang zien we eveneens 2 Vtt, maar dan in fase gedraaid. Op de inverterende ingang meten we 50 mVtt. Daaruit volgt een open-lus-versterking van 80. Dat is niet bijzonder veel, zelfs de simpele 741 haalt ruimschoots 100.000. Op één punt is onze zelfbouw-opamp echter beter: hij levert een rail-to-rail uitgangsspanning van maximaal 9 Vtt.

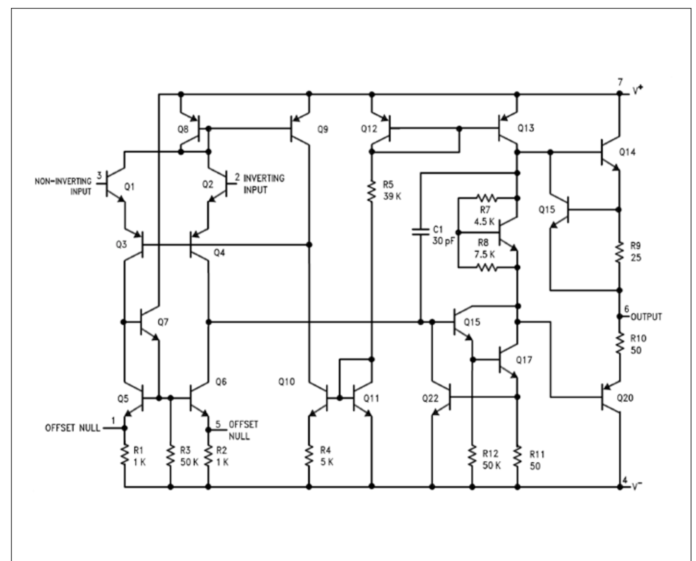
De meeste fabrikanten geven in de datasheet een vereenvoudigde schakeling van het inwendige van de opamp. Een aantal zaken die belangrijk zijn voor de toepassing komen daaruit naar voren. **Figuur 8** is de interne schakeling van de standaard-opamp LM741. U ziet bepaalde overeenkomsten met onze zelfgebouwde opamp met NPN-transistoren aan de ingangen. In de LM741-schakeling zien we een verschilversterker met NPN-transistoren aan de ingang. De emitterstroom kan extern enigszins worden afgeregeld op de beste symmetrie om de offsetspanning op nul af te regelen. Via een stroomspiegel wordt het signaal naar een tussentrap gestuurd, die op zijn beurt een push-pull-eindtrap stuurt. Eén enkele condensator zorgt ervoor dat de interne bandbreedte beperkt blijft tot 1 MHz. Waar de open-lus-versterking tot 10 Hz nog ongeveer 100.000 bedraagt, is dat bij 1 MHz nog slechts éénmaal. De versterking als functie van de frequentie is weergegeven in **figuur 9**, met een logaritmische schaal voor de x-as. Deze afname van de versterking bij hoge frequenties is nodig om in alle gevallen een stabiele versterking te verkrijgen. Het gevolg is echter ook dat een standaard-opamp niet voor heel hoge frequenties gebruikt kan worden.

In het tweede deel van deze miniserie maken we kennis met een ander type opamp (met veldeffect-transistoren in de ingangstrap) die bijzondere toepassingen mogelijk maakt. ◄

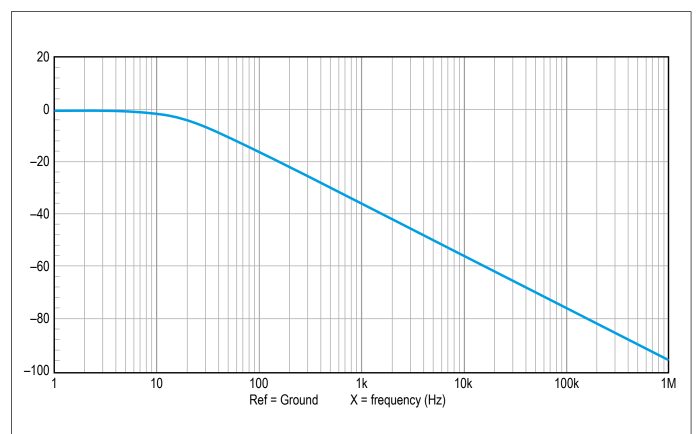
180036-03



Figuur 7. Spanningen in een zelfgebouwde inverterende versterker.



Figuur 8. De LM741 van binnen (afbeelding: Texas Instruments).



Figuur 9. De open-lus-versterking als functie van de frequentie.

eLABS PROJECT

Compacte USB/DMX-Converter

volledig elektrisch geïsoleerd

Jochem Brouwers (Elektor Labs Trainee)

Aan goed geïsoleerde USB-naar-DMX512-converters hangt een professioneel prijskaartje — lees: ze zijn dus erg duur. Daarom heb ik een alternatief voor zelfbouw ontworpen. Het is goedkoper dan professionele converters, maar heeft vergelijkbare specificaties.



Er zijn tegenwoordig veel goedkope converters verkrijgbaar voor het regelen van verlichtingssystemen bij shows, concerten, events, danceparty's en andere evenementen. Helaas zorgen die niet voor een galvanische scheiding tussen de computer en de gevaarlijke hoogspanning op de locatie. Eén van de risico's die dat met zich meebrengt, is de kans dat de USB-poort wordt opgeblazen bij een kortsluiting in een spot of een ander aangesloten DMX-apparaat. Dit zelfbouwproject verslaat de goedkope commerciële 'converters' door **elektrische veiligheid** te bieden.

Hoe het werkt

Het schema in **figuur 1** is tamelijk eenvoudig en is gebaseerd op de Entec Open DMX USB Interface, maar dan voorzien van een volledige elektrische isolatie (zoals we verderop zullen zien). Laten we beginnen met een kleine rondleiding door het schema, één van de laatste *Works of Elektor Art* die zijn geproduceerd door tekenaar Mart Schroijen die in augustus van dit jaar na meer dan 40 dienstjaren bij Elektor met pensioen is gegaan. Een bekend, bijna 'klassiek', IC in de schakeling is natuurlijk de FT232RL (IC1), een USB/seriële UART-interface. Deze implementeert het complete USB-protocol op één chip, er is dus verder geen USB-specifieke hardware meer nodig.

De WPMIB9200501S, IC2, is een DC/DC-converter die is geselecteerd vanwege zijn goede isolatie-eigenschappen, weergegeven door de scheidingslijn die door het schemasymbool loopt. Er zijn dan ook twee gescheiden massalijnen: 'USB-massa' bij de ingang en een ongelabelde massa aan de DMX-kant bij de uitgang.

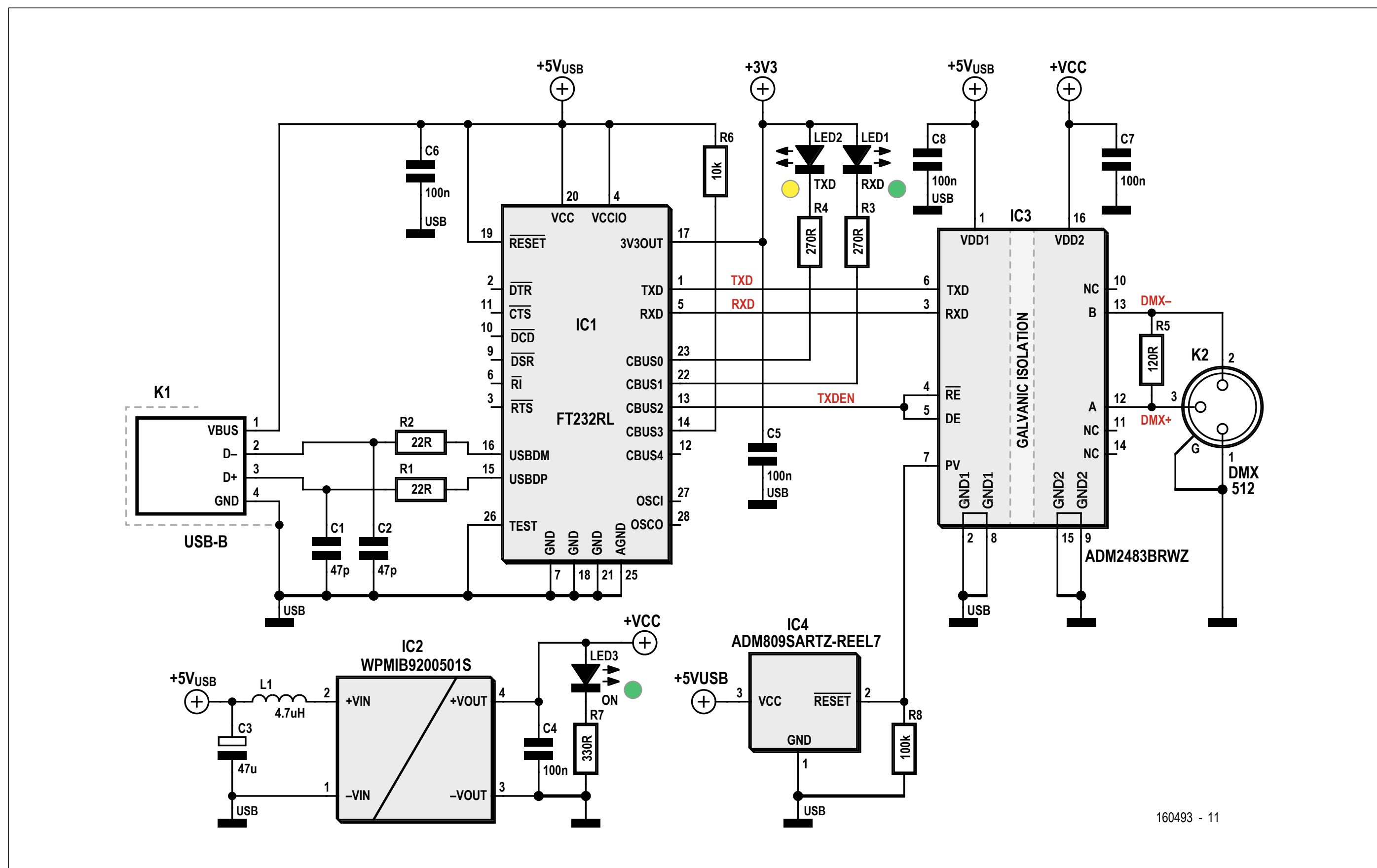
IC4 is een 'microprocessor supervisory circuit'. Er is hier gekozen voor een 809SARTZ-REEL7 van Analog Devices. Deze levert een active-low RESET signaal bij het in- en uitschakelen van de voedingsspanning en bij te lage netspanning. Bij het inschakelen houdt een interne timer RESET-lijn ongeveer 240 ms lang vast.

Dan komen we bij IC3, het hart van de schakeling, een ADM2483BRWZ. Dit is een half–duplex, geïsoleerde RS-485-transceiver, ook weer van Analog Devices. Dit IC behoort tot AD's iCoupler®-familie die specifiek is ontworpen voor het realiseren van elektrische isolatie in digitale systemen. De ADM2483 *differential bus transceiver* is een geïntegreerde, galvanisch geïsoleerde component die is ontworpen voor bidirectionele datacommunicatie over gebalanceerde buslijnen. Hij voldoet aan de standaarden ANSI EIA/TIA-485-A en ISO 8482: 1987(E). Hij bevat een 3-kanaals isolator, een tristate differentiële linedriver en een differentiële input-ontvanger in één behuizing. De digitale kant van de chip kan worden gevoed met 5 V of 3 V. De buskant gebruikt alleen een 5-V-voeding.

De ADM2483 beperkt de stijg- en daaltijden van de signalen om reflecties bij onjuist afgesloten overdrachtslijnen te verminderen. Vanwege deze gereduceerde slew-rate bedraagt de maximale transmissiesnelheid 500 kbps. De ingangsimpedantie is 96 kΩ, zodat er maximaal 256 (!) transceivers op de

Belangrijkste specificaties

- Volledige galvanische scheiding tussen DMX-systeem en laptop
- ADM2483 iCoupler®-geïsoleerde RS485-transceiver
- Compacte IP40-behuizing
- Ondersteunt alle 512 kanalen



Figuur 1: De converter bestaat uit vier IC’s, waarvan er twee, de DC/DC-converter IC2 en de RS485-transceiver IC3, verantwoordelijk zijn voor de volledige galvanische scheiding tussen de DMX-verlichting aan de ene kant en uw kostbare laptop en gezondheid aan de andere kant.



ONDERDELENLIJST

Weerstanden

(alle 5%, 0,1 W, 150 V)
R1,R2 = 22 Ω
R3,R4 = 270 Ω
R5 = 120 Ω
R6 = 10 kΩ
R7 = 330 Ω
R8 = 100 kΩ

Condensatoren

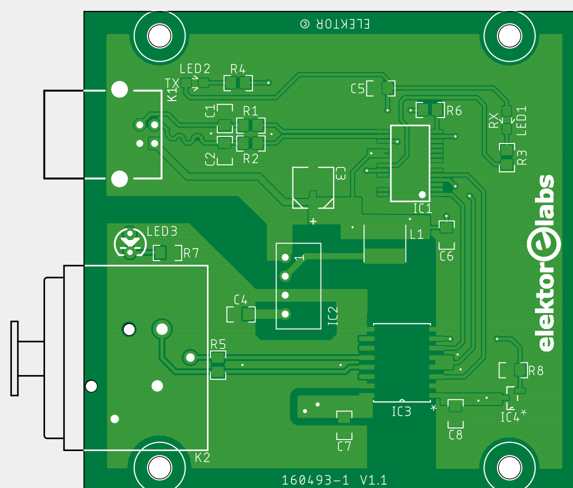
C1,C2 = 47 pF/50 V, C0G/NP0, 0805
C3 = 47 µF/25 V, ESR: 0,36 Ω, SMD, aluminium, elco
C4,C5,C6,C7,C8 = 100 nF/50 V, X7R, 0805

Halfgeleiders

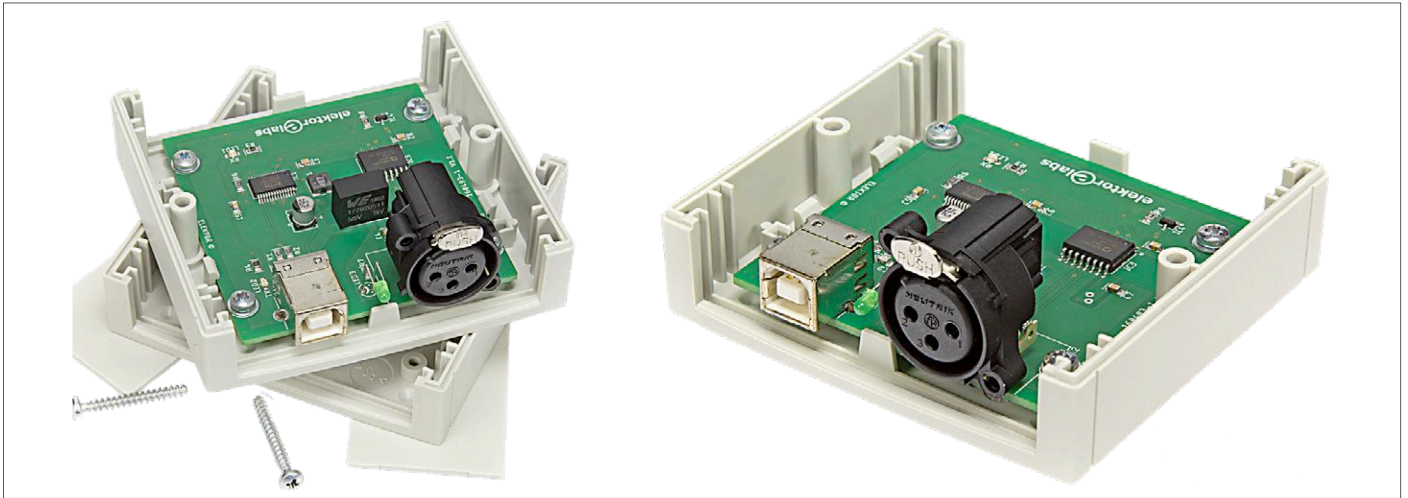
LED1 = groen, 50 mcd, 2,1 V @ 20 mA
LED2 = geel, 50 mcd, 2 V @ 20 mA
LED3 = groen, 3 mm
IC1 = FT232RL
IC2 = WPMIB9200501S
IC3 = ADM2483BRWZ
IC4 = ADM809SARTZ

Diversen

K1 = USB-B-connector
K2 = NC3FAAH2, 3-polig XLR-socket, vergulde contacten, AA-serie
ABS-behuizing, polystyreen, 40 x 85 x 81 mm, IP40, bijv. Bopla type 26085000
M3 x 5 mm kruiskop-schroeven
M3-ring
Print 160493-1 rev.1 (zie ‘In de store’)



Figuur 2: Bij het ontwerp van de print voor de converter stonden compactheid en de elektrische isolatie tussen DMX en USB centraal.



Figuur 3: Twee foto's van het project. De print past perfect in de gespecificeerde Bopla-kast.

bus mogen zitten. De driver heeft een active-high enable-lijn. De differentiële uitgangslijnen van de driver en de differentiële inputs van de ontvanger zijn intern met elkaar verbonden, zodat een differentiële I/O-poort wordt gevormd. Als de driver is uitgeschakeld of wanneer VDD1 of VDD2 = 0 V, vormt hij een verwaarloosbare belasting voor de bus. Verder is er een active-high disable-lijn voor de ontvanger, die de uitgangslijnen van de ontvanger in hoge impedantiemodus zet. Het DMX-systeem waarmee u uw publiek wilt imponeren en verblinden wordt aangesloten op K2, dat is vanzelfsprekend een XLR-connector. De PC wordt aangesloten op de USB-B-connector K1. De massa van het DMX-systeem en de USB-lijn zijn van elkaar gescheiden in IC3, dus heeft het IC daarvoor ook aparte aansluitingen. Let op de twee massa-symbolen in het schema: de ene is de DMX-massa (ongelabeld) en de andere is USB GND. De rest van de schakeling bestaat uit weinig meer dan de verplichte ontkoppelcondensatoren voor de voeding en de indicatie-LED's voor de voeding en de activiteit op de TXD- en RXD-lijnen (respectievelijk LED3, LED2 en LED1) met de gebruikelijke serieweerstanden.

Print en opbouw

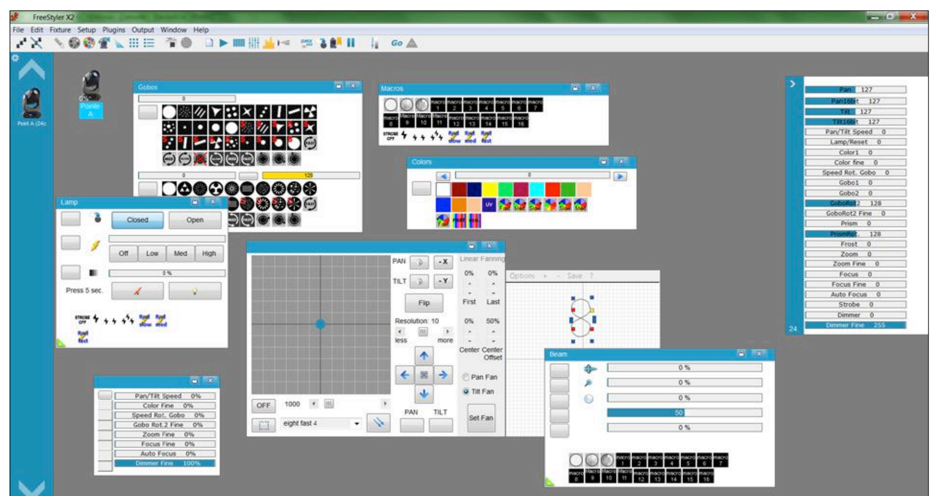
Omdat de schakeling bijna helemaal uit SMD-componenten bestaat, is de converter niet echt gemakkelijk na te bouwen. Toch houden we vast aan de traditie om de print-layout af te drukken (**figuur 2**) en de onderdelenlijst te publiceren. De layout kan ook worden gedownload van de projectpagina bij dit project [1]. Ook kan de kale print 'los' worden besteld. Aan het printontwerp is duidelijk te zien, dat er veel aandacht is besteed aan alle aspecten van elektrische isolatie — het USP (*Unique Selling Point*) van deze converter. Alle printsporen die van belang zijn voor de elektrische isolatie zijn geplaatst op onderlinge afstanden die meer dan voldoende zijn voor de vereiste veiligheidsgraad.

De vorm van print is afgestemd op een Bopla-behuizing van het type 26085000ABS. In de foto's in **figuur 3** zijn de print en de behuizing te zien vanuit verschillende gezichtshoeken.

De praktijk

De converter is getest met Freestyler DMX onder Windows. Een voorbeeld van dat programma in actie is te zien in **figuur 4**. De werking op MacOS is getest met Lightkey en QLC+. Alle 512 kanalen zijn werkelijk te gebruiken, wat ook een voordeel is in vergelijking met goedkope converters. Er zijn diverse tests uitgevoerd met RGB-LED-spots en scanners en alles bleek naar tevredenheid te functioneren. Bovendien was het volkomen veilig!

(180356-04)



Figuur 4: Freestyler DMX is een gratis DMX-besturingsprogramma dat door veel DJ's, VJ's en theaterlichttechnici wordt gebruikt. Freestyler werkt uitstekend met de hier beschreven converter.

Weblink

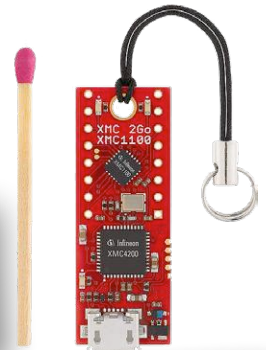
[1] Projectpagina bij dit artikel: www.elektormagazine.nl/180356-04

Doe mee met de

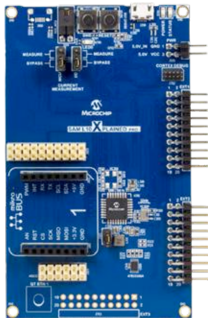
elektor
board games
POWERED BY AVNET SILICA AND EBV ELEKTRONIK

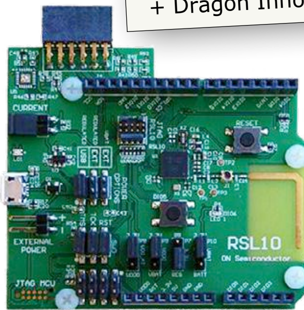
Eerste prijs*
Componenten ter waarde van € 5.000
+ Ondersteuning om uw idee om te zetten in een marktrijp product
+ Dragon Innovation Product Planner Software
+ Uw project presenteren op electronica 2018



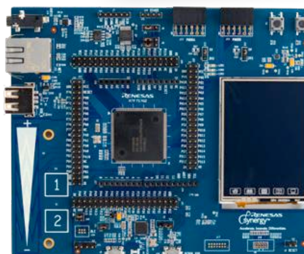


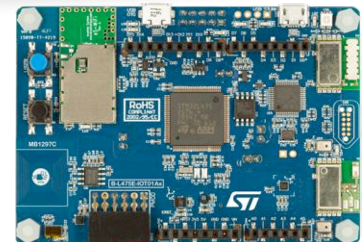
Tweede prijs*
Componenten ter waarde van € 2.500
+ Ondersteuning om uw idee om te zetten in een marktrijp product
+ Dragon Innovation Product Planner Software





Derde prijs*
Ondersteuning om uw idee om te zetten in een marktrijp product
+ Dragon Innovation Product Planner Software





* Avnet Silica en EBV Elektronik kiezen **allebei** een eerste, tweede en derde prijswinnaar, dus er zijn in totaal **zes** prijzen te winnen.

Hoe kunt u meedoen?

- Kies één van de zes beschikbare gratis boards voor uw project;
- Vul het formulier in;
- Dien uw projectvoorstel in ter goedkeuring;
- En voer het uit!

Veel plezier met de Board Games 2018!

Avnet Silica, EBV Elektronik en het Elektor Labs-team

Weblink

[1] Elektor Board Games: www.elektormagazine.nl/widgets/board-games

HOMELAB PROJECT

Magische lamp

multicolour zonder draden

Roel Arits (Nederland)

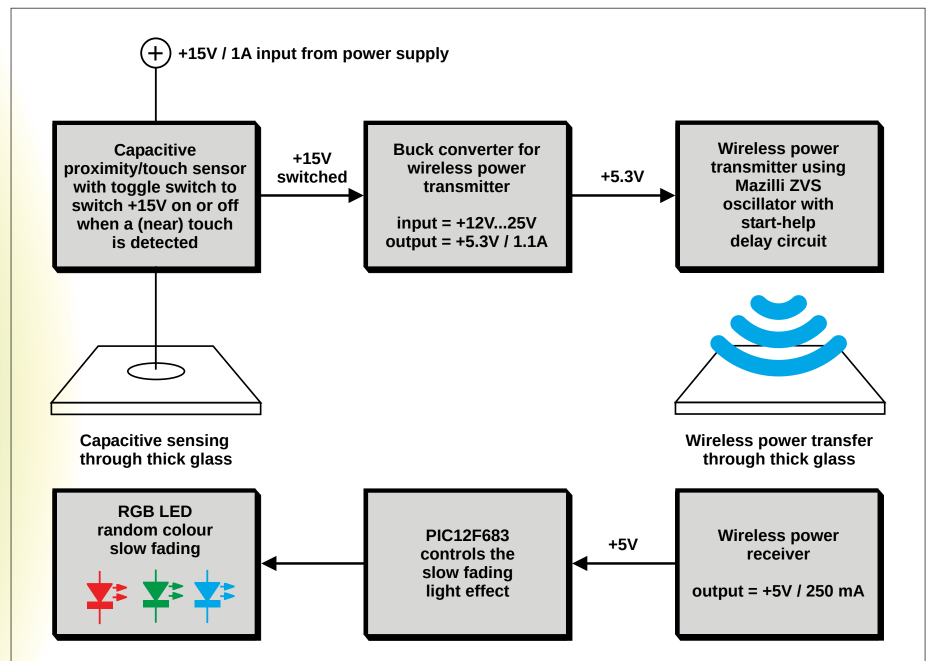
Wat is er mooier dan wanneer twee mensen hun hobby in een gemeenschappelijk project combineren? Inderdaad: wanneer het twee echtelieden betreft. Zo ook in dit geval: de auteur combineerde zijn voorliefde voor LED-verlichtingsprojecten met het artistieke glasmozaiekwerk van zijn echtgenote, resulterend in een unieke en geheel draadloze 'lichtsculptuur'.



De combinatie van RGB-LED's met (transparant) glasmozaiek resulteert in fraaie lichteffecten. Maar het werken met glas vormt voor een elektronicus een ware uitdaging, omdat het niet eenvoudig is gaten (voor de voeding en een aan/uit-schakelaar) in glas te maken. En om de haverklap een batterij vervangen is ook niet zo'n geweldig idee — al was het alleen maar vanwege de milieubelasting. Daar was dus een andere aanpak nodig, en met een beetje elektronisch vernuft is die niet eens zo moeilijk te realiseren. In de eerste plaats is het mogelijk de RGB-LED plus omringende elektronica in het inwendige van de lamp draadloos te voeden, vergelijkbaar met het draadloos opladen van de accu van een handy. Dat dit kan, is te danken aan het betrekkelijk geringe stroomverbruik van

Eigenschappen

- (Pseudo)random-generator stuurt RGB-LED voor wisselende kleuren
- Bediening met capacitieve aanraak/nabijheidsschakelaar
- Draadloze voeding door middel van power-oscillator



Figuur 1. Het blokschema telt veel blokken, maar in de praktijk is de elektronica niet al te ingewikkeld.

een LED plus aansturing (in de vorm van een zuinige microcontroller).

En wanneer we een uiterst gering standby-stroomverbruik voor lief nemen, vormt een aan/uit-schakelaar in de vorm van een capacitieve aanraak/nabijheidsschakelaar een uitstekende oplossing. In de praktijk zijn nauwelijks problemen te verwachten omdat in beide gevallen de te overbruggen afstand slechts een paar centimeter of nog minder bedraagt.

Blokschema

Aan elektronica hebben we dus een capacitieve aan/uit-schakelaar ('toggle switch') nodig die de voedingsspanning voor de energie-'zender' schakelt; en in de lamp zelf een ontvanger voor de uitgezonden energie plus een RGB-LED met aansturing.

Wanneer we dat in de vorm van een blokschema gieten, ziet het resultaat er uit als in **figuur 1**.

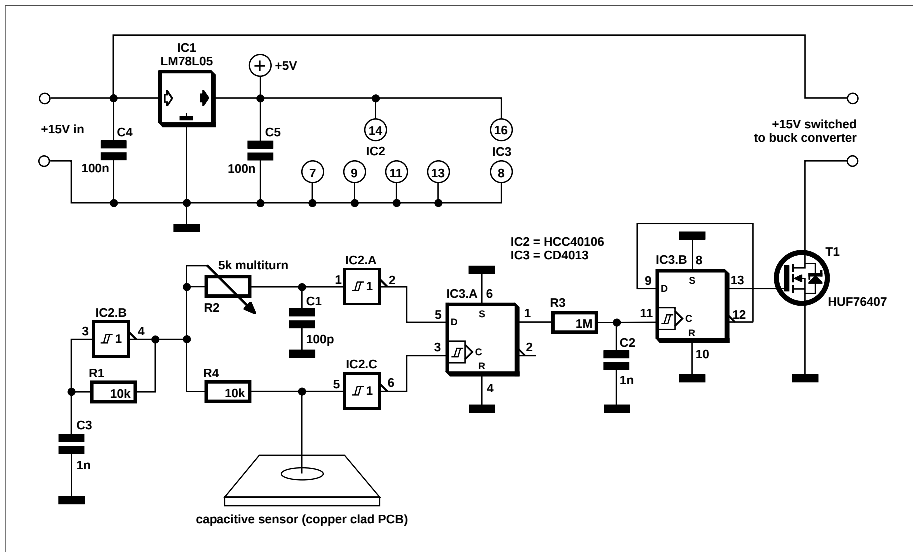
De ingangsspanning van 15 V (bij een maximale stroom van 1 A) wordt geleverd door een standaard netadapter. (Het enige gaatje dat we in de basis van de lichtsculptuur moeten maken is voor de voedingsconnector.) Deze spanning wordt door een capacitieve toggle-switch met behulp van een FET met geringe $Rds_{(on)}$ geschakeld. Daarna komt een buck-converter die een stabiele 5,3 V voor de zender levert.

Voor de draadloze energie-overdracht wordt een Mazilli ZVS-vermogensoscillator gebruikt; omdat deze af en toe startproblemen bleek te geven, is voorzien in een opstartvertraging.

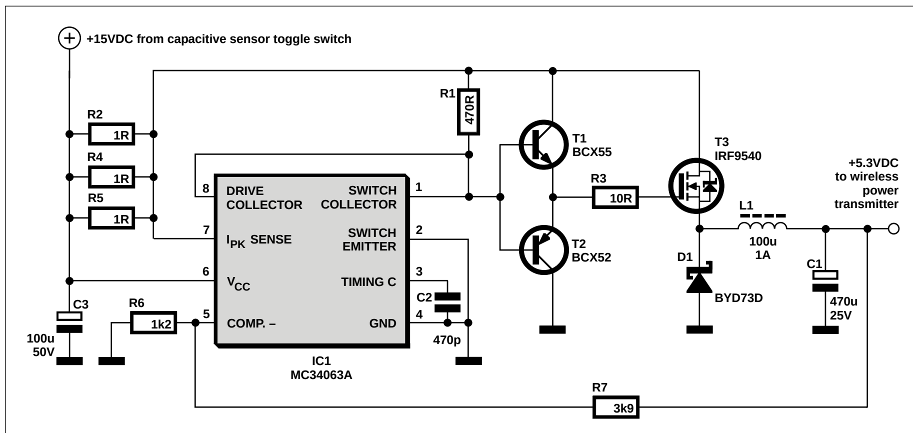
De ontvanger is in wezen een afgestemde kring plus gelijkrichter, gevolgd door een eenvoudige low-drop spanningsregelaar. Deze levert de 5-V-voedingsspanning voor het laatste blok in het blokschema: een kleine PIC-microcontroller die de RGB-LED (of 'losse' LED's) aanstuurt.

Specificaties

- Totaal stroomverbruik ca. 150 mA @ 15 V (oscillator en LED ingeschakeld)
- Zendelektronica verbruikt ongeveer 94 mA @ 15 V
- Ontvanger plus microcontroller/LED verbruiken ongeveer 60 mA @ 5 V
- Standby-stroomverbruik (oscillator en LED uitgeschakeld) ca. 5 µA



Figuur 2. De capacitieve aanraak/nabijheidsschakelaar berust op het meten van een klein capaciteitsverschil.



Figuur 3. Een standaard-applicatie van de MC34063 maakt van de 15 V ingangsspanning een stabiele 5,3 V voor de rest van de elektronica.

Capacitieve schakelaar

We zullen elk blok uit het blokschema nu apart bekijken, te beginnen met de capacitieve schakelaar. Daaraan worden twee belangrijke eisen gesteld: hij moet gevoelig zijn en mag niet door temperatuurverschillen worden beïnvloed — met andere woorden, hij moet stabiel zijn.

Het hart van de schakelaar (waarvan **figuur 2** het schema toont) wordt gevormd door een relaxatie-oscillator [1] rond een CMOS-Schmitt-trigger (IC1B, een van de zes inverterende Schmitt-trigger-buffers in een 40106). Elke zichzelf respecterende electronicus bouwt zo’n oscillator met gesloten ogen op, dus daar gaan we niet verder op in. We volstaan met op te merken dat op de uitgang van IC2B een blokgolf staat met een frequentie van ongeveer 100 kHz en een duty-cycle van 50%.

Het uitgangssignaal van de oscillator gaat naar twee RC-netwerkjes die de blokgolf integreren. Het ene netwerk (bestaande uit R2 en C1) is vast, het andere netwerkje is variabel omdat dit uit R4 en de capacitieve sensor CS (een stuk printplaat) bestaat.

De capaciteit van die sensor zal veranderen (toenemen) wanneer we onze hand erboven houden, waardoor de tijdconstante van het RC-netwerkje zal veranderen (toenemen). Dit veroorzaakt een zekere vertraging ten opzichte van het vaste netwerk. Het verschil in laadtijden (dus die vertraging) is evenredig met de extra capaciteit.

We zouden nu de uitgangen van de beide netwerkjes (die met Schmitt-triggers gebufferd zijn) met een XOR-poortje kunnen vergelijken, maar dat is in de praktijk niet zo’n heel goed idee (we laten het als oefening voor de lezer te beredeneren waarom).

Hier wordt een D-flipflop (IC3A) gebruikt om beide pulstreinen te vergelijken: de flipflop zal alleen pulsen ‘doorlaten’ wanneer het kloksignaal (CLK) vertraagd is ten opzichte van het D-signaal. Als daarentegen het D-signaal vertraagd is, zal de uitgang altijd laag zijn.

R3 en C2 vormen samen een integrator met een zeer grote tijdconstante vergeleken met de breedte van de uitgangspulsen van IC3A.

Flipflop IC3B zal telkens wanneer de spanning over C2 groter wordt dan de drempelspanning van de CLK-ingang omklappen; dat is dus telkens wanneer de sensor voldoende extra capaciteit ziet. Via MOSFET T1 schakelt de flipflop zodoende de 15-V-voedingsspanning van 15 V door naar de converter.

Buck-converter

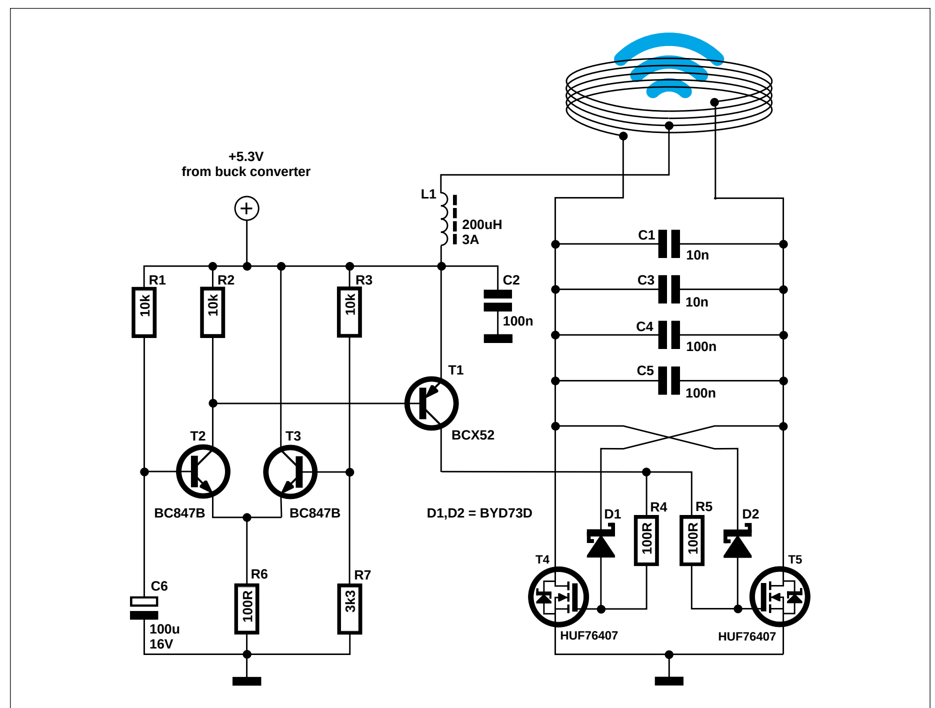
Om de door de netadapter geleverde 15 V terug te brengen naar 5,3 V (wat voldoende is voor de microcontroller plus RGB-LED) gebruiken we (grotendeels) een standaardapplicatie rond de bekende schakelende regelaar MC34063. Voor details over deze chip verwijzen we naar de datasheet [2].

Figuur 3 toont het schema. Het belangrijkste verschil ten opzichte van de standaardapplicatie voor een step-down converter is de uitgangstrap (T1, T2 en FET1). De MC34063 bleek in de praktijk tamelijk warm te worden wanneer deze een continue uitgangsstroom van 1 A moet leveren. Daarom heeft de auteur de uitgangstrap duidelijk overgedimensioneerd om de boel koel te houden (temeer daar de voeding in een geheel gesloten behuizing is gemonteerd en onder normale omstandigheden continu is ingeschakeld), dit onder het motto ‘liever te koel dan te warm’.

Draadloze voeding: de zender

Aan draadloze energie-overdracht is eigenlijk niets geheimzinnigs! Aan de ene kant hebben we een zendspoel en aan de andere kant een ontvangstspoel — hé, wacht even... Twee spoelen? Als die via een ijzerkern magnetisch gekoppeld zijn, hebben we met een doodordinaire transformator te maken! Maar zonder ijzerkern lukt het ook heel behoorlijk, als de afstand tussen de spoelen niet te groot is (enkele centimeters), de frequentie

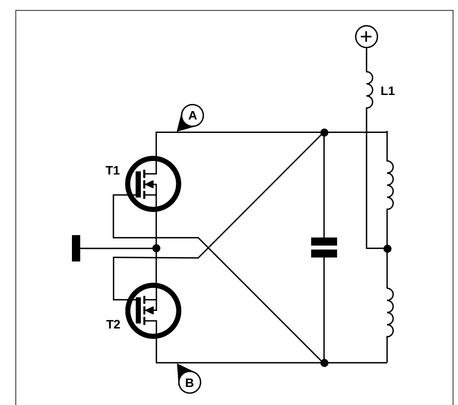
relatief hoog is (een stuk hoger dan bij een nettrafo) en beide spoelen (beter: resonantiekringen) netjes op dezelfde frequentie zijn afgestemd.
Wat we nodig hebben is een oscillator die voldoende vermogen door de zendspoel kan jagen zodat de ontvangstspoel een buikbare hoeveelheid energie kan oppikken.
De auteur heeft gekozen voor een heel simpele oscillator die zonder grote koellichamen veel vermogen kan verwerken: een Mazilli ZVS-oscillator. In **figuur 4** is het complete schema van de zender getekend, en in **figuur 5** hebben we voor een beter begrip de eigenlijke oscillator uitgelicht.
Het belangrijkste deel van deze oscillator is de LC-kring die door de spoel met middenaftakking en de condensator wordt gevormd. Hierdoor heeft de stroom een sinusvormig verloop.
Laten we aannemen dat, na het inschakelen van de voedingsspanning, transistor T1 als eerste in geleiding gaat (geen twee transistoren zijn exact gelijk; één van beide transistoren zal altijd iets sneller inschakelen dan de andere. Er loopt dan door de corresponderende spoelhelft iets meer stroom, waardoor de andere FET minder gatestroom krijgt en zal uitschakelen.
Laten we aannemen dat (in figuur 5) FET T1 als eerste inschakelt. De spanning op punt A zal dan in de buurt van 0 V liggen, terwijl de spanning op punt B een maximum bereikt en dan weer naar nul gaat als de LC-kring een halve cyclus doorloopt.
Op het moment dat de spanning op punt B door nul gaat, krijgt T1 geen gatespanning meer en zal afschakelen. Nu kan de spanning op punt A toenemen zodat T2 zal gaan inschakelen. Deze transistor houdt de spanning op punt B op 0 V zodat T1 uitgeschakeld blijft. Nu voltooit T2 de tweede helft van de cyclus — en zo vervolgens: de oscillator doet waarvoor hij is geschapen en blijft oscilleren. Om te voorkomen dat de oscillator een te grote piekstroom trekt is L1 als smoorspoel toegevoegd.
Het fraaie van deze opzet is dat dit een *zero voltage switching*-oscillator is: de MOSFET's schakelen op het moment dat er (vrijwel) geen spanning over staat. Op die manier worden schakelverliezen geminimaliseerd zodat zelfs bij zeer grote vermogens slechts kleine koellichamen nodig zijn (en in deze specifieke toepassing zelfs helemaal geen).



Figuur 4. De zender bestaat uit een Mazilli ZVS-poweroscillator met inschakelvertraging.

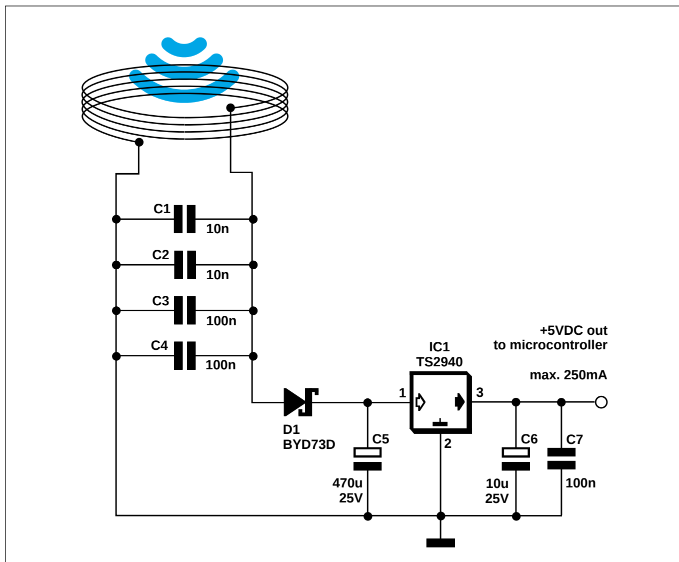
Bij de eerste experimenten met deze schakeling merkte de auteur dat de oscillator niet altijd opstartte. Dat heeft tot gevolg dat één van de FET's in geleiding komt en blijft, terwijl de andere blijft sperren. Na korte tijd resulteert dit in de geur van gesmoorde halfgeleider die iedere elektronicus wel kent — een bepaald onwenselijke situatie. Na enig onderzoek bleek het probleem te worden veroorzaakt door het trage en niet-geleidelijke opkomen van de voedingsspanning van de netadapter.

Er zijn in zo'n geval twee oplossingen denkbaar: het is mogelijk een andere netadapter te zoeken waarbij het probleem niet optreedt — maar dat is uiteraard niet de ideale oplossing. Beter is een 'universele' remedie te vinden die het mogelijk maakt ook langzaam opkomende voedingen te gebruiken.
De auteur heeft het probleem opgelost door een vertraging in te bouwen die ervoor zorgt dat de 100-Ω-gateweerstanden van de MOSFET's ongeveer 250 ms later met de voedingsspanning worden verbonden dan de drains.
De vertraging is opgebouwd rond T1... T3. T2 en T3 vormen een verschilversterker. Na het inschakelen van de voedingsspanning wordt condensator C6 langzaam opgeladen via weerstand R1. Na enige tijd bereikt de spanning over de

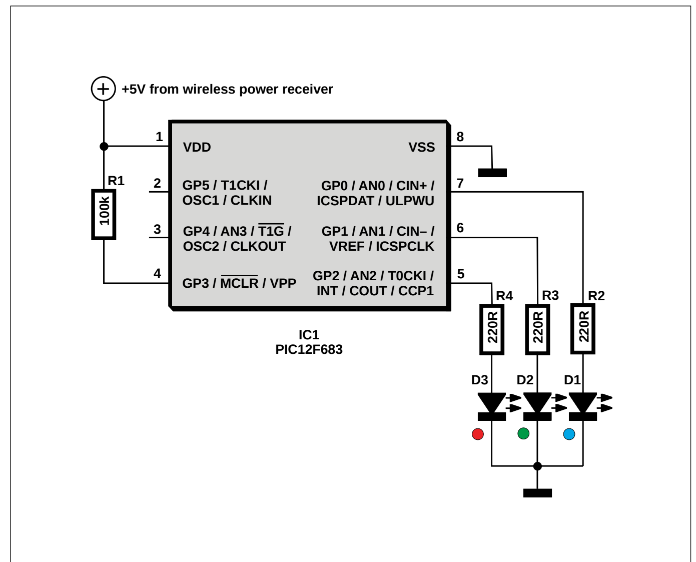


Figuur 5. Aan de hand van dit vereenvoudigde schema kan de werking van de oscillator worden begrepen.

condensator de drempelspanning die met R3/R7 is ingesteld; T2 gaat dan in geleiding en trekt de basis van T1 omlaag. Hierdoor gaat deze transistor geleiden en de beide MOSFET's krijgen via R4 en R5 gatespanning.
Nog enkele details: condensatoren C1 en C3 dienen voor het afstemmen van de LC-kring. Hiermee kan de resonantiefrequentie in geringe mate worden aangepast, mocht dat nodig blijken. De spoel (met middenaftalling) bestaat uit 14 windingen van 1-mm koperlakdraad; de windingen liggen strak tegen elkaar. De hoogte van de spoel (met een diame-



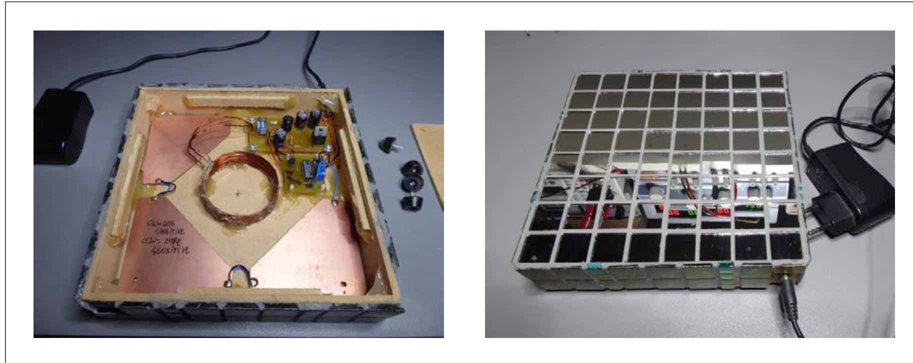
Figuur 6. De ontvanger is de eenvoud zelve.



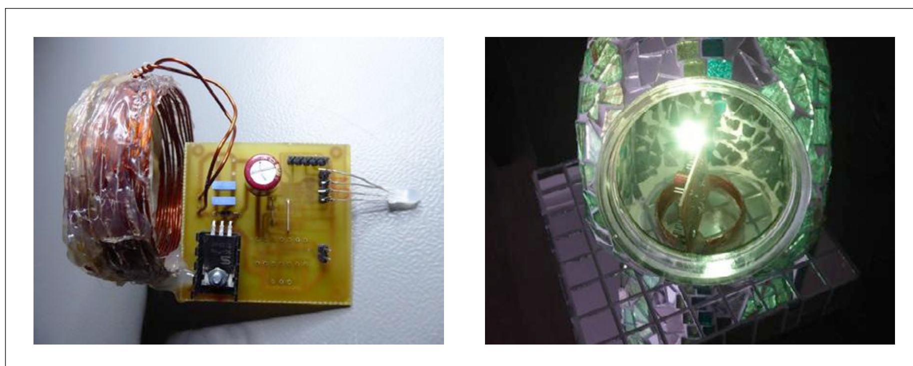
Figuur 7. Er zij licht... Een microcontroller stuurt een RGB-LED aan.

ter van ongeveer 57 mm) bedraagt dan zo'n 16 mm. De zelfinductie bedraagt ongeveer 16 µH.
In combinatie met C4 en C5 (samen 200 nF) heeft de LC-kring een resonantiefrequentie van ongeveer 81 kHz.

Wanneer u deze schakeling wilt nabouwen, maakt het niet uit of de resonantiefrequentie hoger of lager dan deze 81 kHz is, zolang zender en ontvanger maar op dezelfde frequentie zijn afgestemd.



Figuur 8. In de voet van de magische lamp vallen de driehoekige stukken printplaat op die als capacitieve sensor dienst doen.



Figuur 9. De ontvangerspoel plus elektronica los en in de lamp.

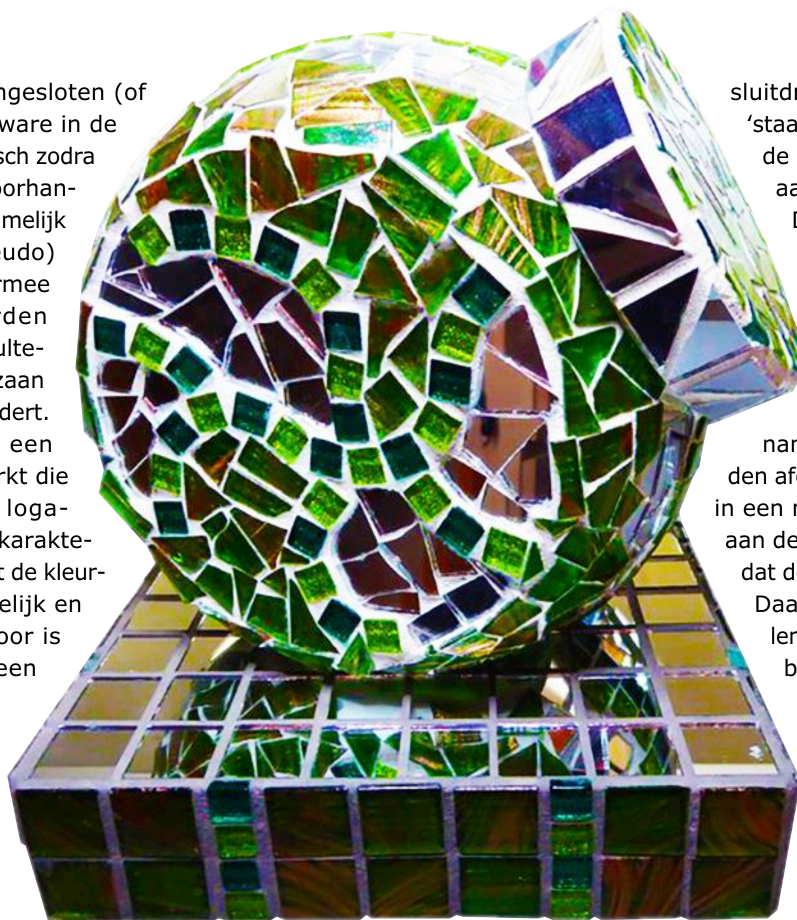
Draadloze voeding: de ontvanger

Het schema van dit deel van de schakeling is getekend in **figuur 6**. We zien hier weer een spoel (die gelijk is aan die van de zender, maar dan zonder middenaftakking) die samen met condensatoren C1...C4 een afgestemde kring vormt (de resonantiefrequentie bedraagt ook hier ongeveer 81 kHz; met C1 en C2 zijn kleine aanpassingen mogelijk). Diode D1 dient als gelijkrichter.
Wacht even — deze opzet doet toch wel heel sterk denken aan een beroemde/beruchte schakeling uit de jaren '60, die bij oudere elektronici ongetwijfeld bekend zal zijn: de jampot-ontvanger [3] van Dr. Blan... Met een op een closetrol gewikkelde spoel, een (afstem)condensator, een diode en een kristal-oortelefoontje was het zonder enig actief onderdeel mogelijk sterke middengolfzenders te ontvangen!
IC1 is een ultra-low-drop spanningsregelaar (met een spanningsval van slechts 0,6 V bij 1 A) die een stabiele voedingsspanning voor de microcontroller en de LED levert.

En tenslotte de controller...

Over de feitelijke lichtbron, dus de controller plus LED's, kunnen we kort zijn — zo kort als het schema klein is (zie **figuur 7**). Het hart wordt gevormd door een kleine (achtbenige) PIC12F683-microcontroller. Aan de ene kant wordt deze gevoed met de 5 V die van de ontvanger afkomstig is, en aan de andere

kant zijn drie LED's aangesloten (of één RGB-LED). De firmware in de controller start automatisch zodra de voedingsspanning voorhanden is, en bestaat voornamelijk uit een zeer trage (pseudo) random-generator waarmee de drie LED's zo worden aangestuurd dat de resulterende kleur uiterst langzaan en onvoorspelbaar verandert. In de firmware is ook een soort linearisatie verwerkt die compenseert voor de logaritmische gevoeligheidskarakteristiek van ons oog, zodat de kleurwisselingen heel geleidelijk en lineair verlopen. Hiervoor is gebruik gemaakt van een look-up tabel.
De firmware (hex-bestand) kan gratis van de projectpagina bij dit artikel [4] worden gedownload.



Figuur 10. De compleet gemonteerde lamp.

Bouw en afregeling

Figuur 8 toont de voet van de magische lamp in gesloten en geopende toestand. De beide printjes met de capacitieve sensor en de vermogensoscillator zijn duidelijk te herkennen, evenals de zendspoel. Ook is goed te zien dat de feitelijke sensor uit drie onderling verbonden relatief grote driehoekige stukken printplaat bestaat. De enige opening is voor de voedingsconnector ten behoeve van de netadapter.
De sensor is bijzonder gevoelig en reageert al op de kleinste capaciteitsverandering. Dat betekent dat wanneer u de massaklem van de probe van uw oscilloscoop aansluit, de totale capaciteit die de sensor 'voelt' al verandert. En hetzelfde gebeurt wanneer u de schakeling op een geaarde labvoeding aansluit.
Daarom is het belangrijk dat instelpotmeter R2 wordt afgeregeld in de situatie waarin de lamp ook daadwerkelijk gebruikt zal worden — dat wil zeggen zonder aangesloten meetinstrumenten.
Het verdient daarom aanbeveling om voor de afregeling een LED op de uitgang van IC3A aan te sluiten, zodat u kunt zien wat u doet.
Wanneer u de lampvoet na afregeling verplaatst (op een tafel die van een ander materiaal is vervaardigd, of naar een plek dichter bij een muur of zo) zal de sensorcapaciteit weer veranderen en zult u R2 opnieuw moeten afregelen. Het is zelfs mogelijk dat de capaciteit wordt beïnvloed wanneer u het netsnoer van de adapter verlegt. Het is maar dat u het weet...
In **figuur 9** ziet u de print van de ontvanger plus microcontroller (die aan de onderzijde van de dubbelzijdige print is gemonteerd) los en in de lamp. Duidelijk is te zien dat de print zonder verdere schroefbevestiging op de aansluitdraden van de ontvangerspoel 'staat'. Wat het programmeren van de microcontroller betreft: via de aansluitingen ICSPCLK en ICSPDAT is dat in-circuit mogelijk. Daartoe verwijzen we naar de datasheet van deze controller [5].
Met de beide condensatoren van 10 nF bij zend- en ontvangstkring kunnen de resonantiefrequenties op elkaar worden afgestemd. De auteur slaagde er in een maximumstroom van 250 mA aan de ontvanger te ontlokken voordat de spanning begon in te zakken. Daarbij bevonden de beide spoelen zich een paar centimeter recht boven elkaar. De vermogensoverdracht kan waarschijnlijk met exacte afstemming van de frequenties nog wat verder worden opgevoerd.
Tenslotte toont **figuur 10** de compleet gemonteerde lamp. De 'magische lamp' is inmiddels al meerdere jaren probleemloos in continubedrijf.
Vooral in het begin bleek de lamp zich 's nachts spontaan in te schakelen; de auteur dacht in eerste instantie aan een storing en wilde de elektronica al demonteren om die nog eens goed tegen het licht te houden — toen bleek dat een nieuwsgierige kat bij het onderzoeken van de nieuwe aanwinst in de huiskamer de capacitieve schakelaar activeerde. Dat toont aan hoe gevoelig de schakeling is... ◄

(170463-01)

Weblinks

[1] Relaxatie-oscillator: https://nl.wikipedia.org/wiki/Oscillator_(elektronica)

[2] Datasheet MC34063: www.onsemi.com/pub/Collateral/MC34063A-D.pdf

[3] Jampot-ontvanger: www.rotterdamsradiomuseum.nl/de-afdelingen/radio-en-elektronica/radio-techniek/jampot-ontvanger/

[4] Projectpagina bij dit artikel: www.elektormagazine.nl/170463-01

[5] Datasheet PIC12F683: ww1.microchip.com/downloads/en/DeviceDoc/41211D_.pdf

Ontwikkelboards van Arrow

u kunt ze gratis krijgen...

Clemens Valens (Elektor Labs)

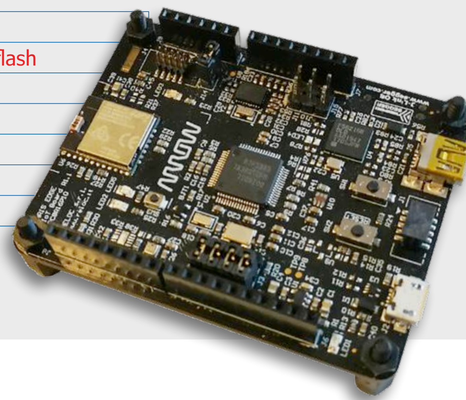
Voor het succes van een nieuw product is het cruciaal dat het als eerste op de markt verschijnt en de nieuwste functies biedt. Maar hoe krijg je dat voor elkaar? Arrow, wereldwijd een van de grootste distributeurs van elektronische onderdelen, ondersteunt startups en development engineers met een reeks tools die de ontwikkeling van producten vergemakkelijkt en een snelle marktintroductie mogelijk maakt.

Het snel realiseren van werkende prototypes wordt door Arrow ondersteund met een uitgebreide reeks ontwikkelboards voor uiteenlopende toepassingen, waarbij het crowdfunding-platform Indiegogo beschikbaar is om het product te lanceren. In dit artikel wordt een selectie van de populairste 'dev'-boards van Arrow gepresenteerd. Bedenk hierbij dat dit slechts een selectie is – u vindt er nog veel meer onder 'Arrow Development Tools' op de website van de distributeur. Let op: deze boards kunnen gratis worden verkregen. Als u een levensvatbaar project aan Arrow presenteert, zullen zij u met gratis hardware en support helpen uw droom te verwezenlijken. Voor meer informatie: **www.elektormagazine.nl/arrow-dev-boards** .

ARIS Edge S1 en Aris Edge S3

De ARIS Edge en Edge S3 zijn edge-nodes voor het Internet of Things (IoT). Ze zijn gebaseerd op een Renesas Synergy MCU (S1 voor de Edge S1, S3A3 voor de Edge S3) en bevatten veel functies die ze geschikt maken voor 'smart-sensing' en het IoT. Voor de communicatie met andere apparaten en de cloud is een flexibele multiprotocol-radiomodule beschikbaar die Bluetooth Low Energy (BLE), Thread en Zigbee-stacks ondersteunt. De sensorfuncties van het board omvatten temperatuur, relatieve vochtigheid, luchtdruk en omgevingslichtniveau, alsmede bewegingsdetectie dankzij een traagheids-meeteenheid (IMU) met 9 vrijheidsgraden (DOF) en sensor-fusion mogelijkheden.

Brein Edge S1	Renesas Synergy S1 ARM Cortex-M0+, 32 MHz, 16 KB RAM & 128 KB flash
Brein Edge S3	Renesas Synergy S3A3 ARM Cortex-M4 met FPU, 48 MHz, 96 KB RAM & 512 KB flash
Connectiviteit	USB, multiprotocol-radio (BLE, Thread en Zigbee)
Sensoren	Vochtigheid, temperatuur, 9-DOF traagheids-meeteenheid (IMU)
Uitbreidingen	Arduino shield-connectoren, I²C-connector, LCD, resistief aanraakscherm
Programmeren	JTAG, JLink (+ JTAG voor radiomodule)
Voeding	USB, knoopcel



Meer informatie

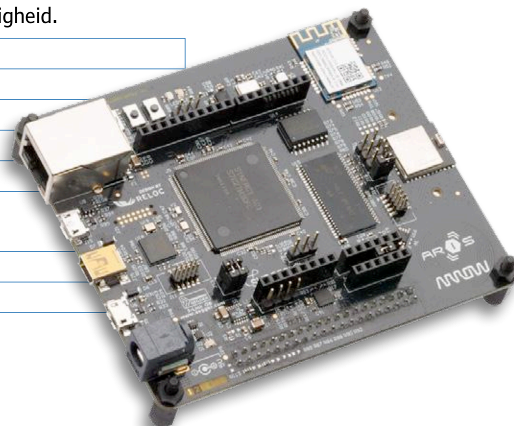
ARIS Edge S1: www.arrow.com/en/products/aris-edge/arrow-development-tools

ARIS Edge S3: www.arrow.com/en/products/aris-edge-s3/arrow-development-tools

ARIS IoT board (BLE) en ARIS gateway (maas)

De ARIS IoT- en gateway-boards zijn kant-en-klare IoT-platforms die gebruik maken van het Renesas Synergy development-framework. Deze boards bevatten een Renesas Synergy S7 MCU met 240-MHz ARM Cortex-M4-kern, beschikken over veel IoT-functies zoals hubs, gateways en maasnetwerken, maar kunnen ook als edge-node worden gebruikt. Voor de communicatie met andere apparaten en de cloud zijn WiFi en Ethernet 10/100 beschikbaar. Het ARIS IoT-board is uitgerust met een Bluetooth Low Energy (BLE 4.1/4.2) module; de ARIS gateway heeft een maas-netwerkmodule voor Thread en Zigbee. Beide boards hebben een NFC tag en een crypto-bootloader, en ondersteunen draadloze (OTA) firmware-updates. Het on-board sensorpakket bestaat uit een drie-assige versnellingsmeter, een twee-assige gyroscoop en sensoren voor temperatuur en vochtigheid.

Brein	Renesas Synergy S7 Cortex-M4 met FPU, 640 KB RAM & 4 MB flash
On-board geheugen	32 MB SDRAM, 64 MB QSPI flash geheugen, microSD
Connectiviteit	USB, Ethernet 10/100, WiFi b/g/n, NFC Forum Type 2 tag
IoT-board	Bluetooth 4.1/4.2, Bluetooth 5 beacon-functie
Gateway-board	Zigbee en Thread radio 2.4 GHz IEEE 802.15.4
Sensoren	Vochtigheid, temperatuur, twee-assige gyroscoop, drie-assige versnellingsmeter
Uitbreidingen	Arduino shield-connectoren, I²C-connector, LCD, resistief aanraakscherm
Voeding	DC-connector, USB mini & micro



Meer informatie

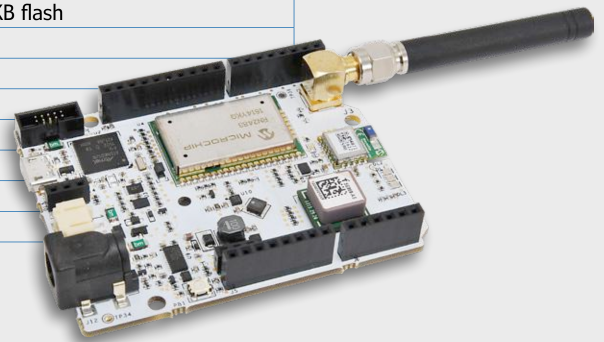
ARIS IoT Board: www.arrow.com/en/products/aris/arrow-development-tools

ARIS IoT Gateway: www.arrow.com/en/products/aris-gateway/arrow-development-tools

Lion (LoRa) en Fox (Sigfox)

Een SAMD21 32-bits ARM Cortex-M0+ ultra low-power microcontroller van Microchip vormt het hart van de Lion- en Fox-boards uit de SmartEverything-reeks. Daarnaast vinden we een LoRa (Lion) of Sigfox (Fox) module, GPS en Bluetooth Low Energy (BLE) modules en een crypto-authenticatie chipset. De boards hebben de Arduino Uno-vormfactor en worden ondersteund door de Arduino-IDE voor snelle en eenvoudige software-ontwikkeling. Omdat er naast de Arduino bootloader een SWD-poort voor programmeren en debuggen aanwezig is, kan ook Atmel Studio worden gebruikt.

Brein	Microchip SAMD21 ARM Cortex-M0+, 48 MHz, 32 KB RAM & 256 KB flash
On-board geheugen	32 KB EEPROM
Connectiviteit	USB, Bluetooth BLE
Lion-board	LoRa
Fox-board	Sigfox
Sensors	GPS
Uitbreidingen	Arduino Uno shield-connectoren
Voeding	DC-connector, USB, Li-Po-accu (>= 700 mAh)



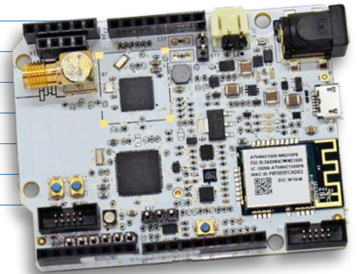
Meer informatie

MC27561-Lion: www.arrow.com/en/products/mc27561-lion/arrow-development-tools
MCS7561-Fox: www.arrow.com/en/products/mcs7561-fox/arrow-development-tools

Tiger

Het Tiger-board heeft twee processoren: een Microchip SAMD21 en een NXP KW41. Hoewel de KW41 is bedoeld voor multi-protocol radiocommunicatie via Bluetooth BLE 4.2, Thread en Zigbee, kan deze ook als hoofd-microcontroller worden geprogrammeerd. Dankzij de ingebouwde arbiter kunnen beide MCU's met de WiFi-module werken en via de seriële poort communiceren. Net als de Lion- en Fox-boards heeft de Tiger de Arduino Uno-vormfactor, en wordt hij ondersteund door de Arduino-IDE voor snelle en eenvoudige software-ontwikkeling. Omdat er naast de Arduino-bootloader een SWD-poort voor programmeren en debuggen aanwezig is, kan ook Atmel Studio worden gebruikt.

Brein 1	Microchip SAMD21J18A-64 ARM Cortex-M0+, 48 MHz, 32 KB RAM & 256 KB flash
Brein 2	NXP KW41 ARM Cortex-M0+, 48 MHz, 128 KB RAM & 512 KB flash
On-board geheugen	32 KB I²C EEPROM, 4 Mb SPI EEPROM
Connectiviteit	USB, WiFi, multi-protocol radio (BLE 4.2, Thread & Zigbee)
Uitbreidingen	Arduino Uno shield-connectoren, FTDI-connector
Voeding	DC-connector, USB, Li-Po-accu (>= 700 mAh)



Meer informatie

MC27561-Tiger: www.arrow.com/en/products/mc27561-tiger/arrow-development-tools

Dragonfly

Het Dragonfly-board is compatibel met Arduino Zero en bevat drie draadloze communicatie-opties — Dusty, Sigfox en WiFi — voor korte en lange afstanden. Met dit board kan het Dusty ecosysteem (gebaseerd op SmartMesh IP) direct of via een SmartMesh IP WiFi-gateway met de cloud worden gekoppeld. De Dragonfly heeft de Arduino Zero-vormfactor, en wordt ondersteund door de Arduino-IDE voor snelle en eenvoudige software-ontwikkeling. Omdat er een SWD-poort beschikbaar is voor programmeren en debuggen kunnen ook andere ARM-programmeeromgevingen worden gebruikt.

Brein	Microchip SAMD21J18B-MU ARM Cortex-M0+, 48 MHz, 32 KB RAM & 256 KB flash
On-board geheugen	32 KB I²C EEPROM
Connectiviteit	USB, WiFi b/g/n, Dusty SmartMesh IP, Sigfox
Uitbreidingen	Arduino Zero shield-connectoren, standaard FTDI-console poort
Voeding	DC-connector, USB, 3-V-batterij (bijv. 2 × AA)



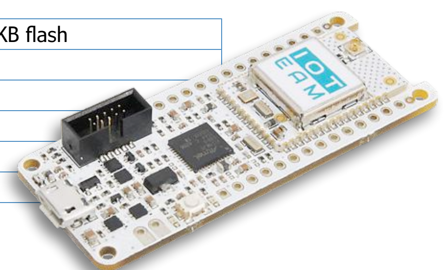
Meer informatie

MC27561-Dragonfly: www.arrow.com/en/products/mc27561-dragonfly/arrow-development-tools

Dustino

Dustino is een klein board voor Dusty (SmartMesh IP) maasnetwerk-toepassingen. Het is compatibel met de Arduino MKR1000 en heeft dezelfde vormfactor. Er zijn twee uitvoeringen: met een printantenne (Dustino ITM-DOPA-B-01) of met een U.FL-connector (Dustino ITM-DOUF-B-01). SmartMesh IP-netwerken zorgen voor meer dan 99,999% betrouwbare data in industrieel IoT bij een batterijlevensduur van meer dan 10 jaar. Als afgeleide van de Arduino MKR1000 wordt Dustino volledig ondersteund door de Arduino IDE. Omdat een SWD-poort beschikbaar is voor programmeren en debuggen kunnen ook andere ARM-programmeeromgevingen worden gebruikt.

Brein	Microchip SAMD21G18A-48 ARM Cortex-M0+, 32 MHz, 32 KB RAM & 256 KB flash
Connectiviteit	USB, Dusty SmartMesh IP
Dustino ITM-DOPA-B-01	Printantenne
Dustino ITM-DOUF-B-01	U.FL-connector
Uitbreidingen	Arduino MKR1000-connectoren
Voeding	DC-connector, USB, 3-V-batterij (bijv. 2 x AA)



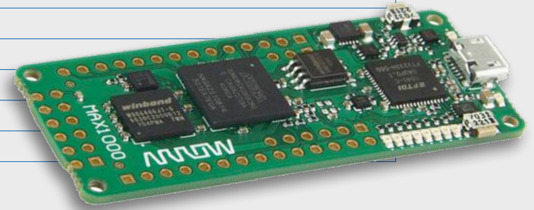
Meer informatie

Dustino ITM-DOPA-B-01: www.arrow.com/en/products/itm-dopa-b-01/arrow-development-tools
Dustino ITM-DOUF-B-01: www.arrow.com/en/products/itm-douf-b-01/arrow-development-tool

MAX1000

Dat FPGA's ook meedoen bij IoT- en snelle prototype-ontwikkeling bewijst het MAX1000-board. Het is gebaseerd op een MAX10 FPGA van Intel en beschikt over 8K logische elementen, 378 Kb RAM, 1.376 Kb Flashgeheugen, een 18 × 18 bits vermenigvuldiger en een ADC. Er kan een 32-bits Nios II softcore microcontroller met I^{2}C- of SPI-poort op draaien, waardoor het toegankelijk wordt voor FPGA-nieuwkomers, zonder de experts tekort te doen. De MAX1000 wordt ondersteund door de gratis Quartus Prime Lite ontwikkelomgeving. Het board kan via JTAG of de on-board Arrow USB Blaster worden geprogrammeerd.

Brein	Intel MAX10 10M08SAU169C8G, 8K LE, 4,75 KB RAM, 172 KB Flash (geschikt voor Nios II 32-bits MCU softcore)
On-board geheugen	64 Mb SDRAM, 64 Mb Flash
Connectiviteit	USB
Sensor	3-assige versnellingsmeter
Uitbreidingen	Arduino MKR-connectoren, PMOD, gebruikers-I/O
Programmeren	JTAG, USB Blaster
Voeding	USB



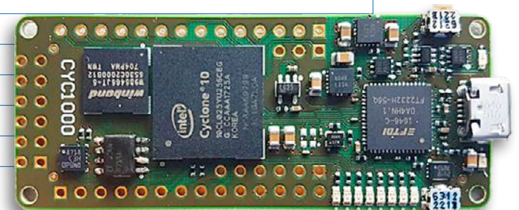
Meer informatie

MAX1000: www.arrow.com/en/products/max1000/arrow-development-tools

CYC1000

Het CYC1000-board lijkt op het MAX1000-board, maar is toch anders. Dit komt door de Cyclone 10 FPGA van Intel die is uitgerust met 24K logische elementen (LE), 594 Kb RAM, en een 18 × 18 bits vermenigvuldiger (of twee 9 × 9 bit vermenigvuldigers). Er is een volledige reeks DSP-IP's voor algoritmische versnelling beschikbaar waarmee dit board een zwaargewicht voor signaalverwerking wordt. Uiteraard kan er een 32-bits Nios II softcore-microcontroller met I^{2}C- of SPI-poort op draaien. De CYC1000 wordt ondersteund door de gratis Quartus Prime Lite ontwikkelomgeving. Het board kan via JTAG of de on-board Arrow USB Blaster worden geprogrammeerd.

Brein	Intel Cyclone 10 LP 10CL025YU256C8G, 24K LE, 74,25 KB RAM (geschikt voor Nios II 32-bits MCU softcore)
On-board geheugen	64 Mb SDRAM, 16 Mb Flash
Connectiviteit	USB
Sensor	3-assige versnallingsmetermeter
Uitbreidingen	Arduino MKR-connectoren, PMOD, gebruikers-I/O
Programmeren	JTAG, USB Blaster
Voeding	USB



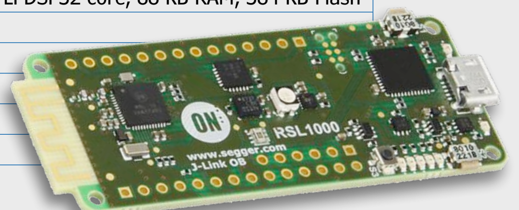
Meer informatie

CYC1000: www.arrow.com/en/products/cyc1000/arrow-development-tools

RSL1000

RSL1000 is een ontwikkelboard voor het IoT dat is gebaseerd op de nieuwste RSL10 Bluetooth 5.0 multiprotocol-radio System on Chip (SoC) van ON Semiconductor. Met zijn standaard Arduino MKR-vormfactor is het bord ideaal voor de ontwikkeling van IoT-edge-nodes, prototypes voor wearables of toepassingen met Bluetooth 5.0.

Brein	ON Semiconductor RSL10 multiprotocol-radio SoC, ARM Cortex-M3 core + LPDSP32 core, 88 KB RAM, 384 KB Flash
Connectiviteit	USB, Bluetooth 5.0
Sensoren	nog te bepalen
Uitbreidingen	Arduino MKR-connectoren
Programmeren	J-Link OB
Voeding	USB



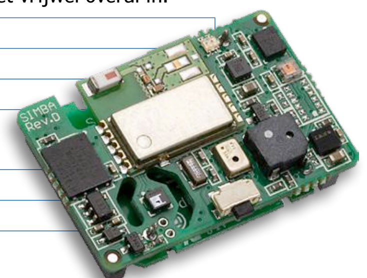
Meer informatie

RSL1000: www.arrow.com/rsl1000

Simba-Pro

Als afmetingen en energieverbruik van belang zijn, is Simba-Pro het board dat u nodig hebt. Het is het basisboard uit de *Sensi*BLE IoT system-on-board (SoB) reeks met Bluetooth (BLE) en acht sensoren. Het bevat een ARM Cortex-M4 32-bits low-power microcontroller van ST, een rood/groene LED, een gebruikers-druktoets en een buzzer. Er is zelfs een AT25XE041B-flashgeheugen van 4 Mb voor dataopslag. Het board wordt gevoed uit een knoopcel, werkt nog bij 2 V en communiceert met het IBM Watson IoT cloud-platform. Met afmetingen van slechts 20 × 30 mm past het vrijwel overal in.

Brein	STmicroelectronics STM32L476RG ARM Cortex-M4, 80 MHz, 128 KB SRAM, 1 MB Flash
On-board geheugen	4 Mb Flash
Connectiviteit	Bluetooth BLE 4.1, USB-voorbereid
Sensors	Temperatuur, vochtigheid, 3-assige versnellingsmeter, 3-assige gyroscoop, 3-assige magnetometer, luchtdruk, microfoon, licht (kleur, IR, omgeving)
Uitbreidingen	13-pens connector, 14-pens connector
Voeding	4,5-7 VDC, 2-3,6 V batterij (CR2025; CR2032)



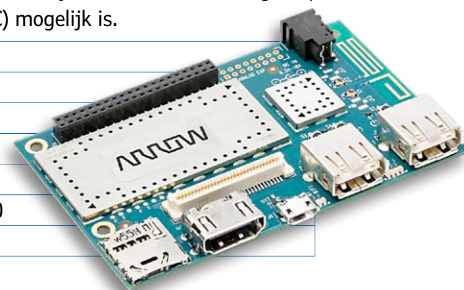
Meer informatie

Simba-Pro: www.arrow.com/en/products/simba-pro/sensiedge

Dragonboard 410c

Het DragonBoard 410c bevat de Qualcomm Snapdragon 410 processor, een quad-core ARM Cortex-A53, met een 1,2-GHz-klok per kern, die geschikt is voor 32-bits en 64-bits functies. Het board ondersteunt besturingssystemen zoals Android 5.1, Linux en Windows 10 IoT core. Het DragonBoard voldoet aan de 96Boards Consumer Edition specificatie. Naast WiFi, Bluetooth en GPS is het ook geschikt voor multimedia, dankzij de GPU en de beeldsignaalprocessor (ISP) met camera-ondersteuning tot 13 MP, waardoor 1080p HD videoweergave en opname met H.264 (AVC) mogelijk is.

Brein	Qualcomm Snapdragon 410 quad-core ARM Cortex-A53
On-board geheugen	1 GB LPDDR3 533 MHz, 8 GB e.MMC 4.51, SD 3.0 (UHS-I)
Connectiviteit	USB, WiFi b/g/n, Bluetooth 4.1
Multimedia	HDMI (audio & video), camera ondersteuning (13 MP)
Sensoren	GPS
Uitbreidingen	40-pens low-speed (LS) & 60-pens high-speed (HS) connectoren, USB-A, USB 2.0
Voeding	6,5...18 (12) V_{DC}



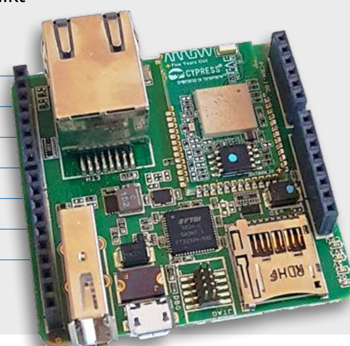
Meer informatie

Dragonboard 410c: www.arrow.com/en/products/dragonboard410ciotsdk/arrow-development-tools

Quadro IoT WiFi Kit

Het Quadro-board met de afmetingen van een Arduino-Uno bevat een afzonderlijke kleine (18 × 30 mm) print met een hooggeïntegreerde system-on-chip (SoC) van Murata. Er zijn twee SoC-versies: de 1GC en de 1GC-Imp05. Beide bevatten een Cypress CYW43907 met een ARM Cortex-R4 kern, 2 MB SRAM, een 2,4 GHz & 5 GHz WiFi/Ethernet-controller en daarnaast een crypto-engine en HF-elektronica. De 1GC-Imp05 is geschikt voor Electric Imp, een hard- en software-platform voor beveiligd IoT. De kleine print is een productierijpe, volledig gecertificeerde reken- en communicatiemodule.

Brein	Cypress CYW43907 ARM Cortex-R4, 320 MHz, 2 MB RAM
On-board geheugen	128 MB Flash, microSD card
Connectiviteit	USB, Ethernet 10/100, dual-band WiFi a/b/g/n
Uitbreidingen	Arduino shield-connectoren
Programmeren	JTAG
Voeding	USB



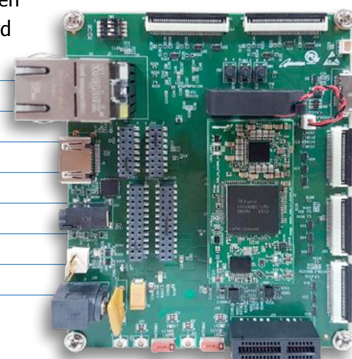
Meer informatie

Quadro IoT WiFi Kit: www.arrow.com/en/products/sh-pcbm-1gc/arrow-development-tools

EIC-Q820-210

De EIC-Q820-210 ontwikkelkit is gebaseerd op de Eragon 820 System-on-Module (SoM) met een Qualcomm Snapdragon 820 SoC. Het board is bedoeld voor het snel realiseren van toepassingen met 4K video, connectiviteit, high-end graphics, energie- en accu-efficiëncy en een flinke rekenkracht, waardoor het een goede keus is voor 4K-cameras, displays, UAV's en drones. Dit krachtige board is geschikt voor Android Lollipop 5.1.1, Debian-8.0 Linux en Windows 10.

Brein	Qualcomm Snapdragon 820 64-bits CPU + GPU + DSP
On-board geheugen	4 GB LPDDR4, 32 GB UFS / 16 GB eMMC
Connectiviteit	USB, Ethernet 10/100/1000, WiFi a/b/g/n, Bluetooth 4.1 + BLE
Sensoren	GPS, versnellingsmeter, gyroscoop, magnetometer
Uitbreidingen	Audio, camera, PCIe WiGig, PCIe SATA, 4K HDMI, low-speed uitbreidingsconnector
Programmeren	JTAG
Voeding	DC-connector 12 V



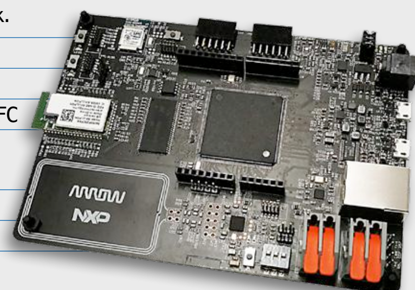
Meer informatie

EIC-Q820-210: www.arrow.com/en/products/eic-q820-210/einfochips-limited

HMI Arrow NXP IoT board (HANI)

Dit board is gericht op HMI (Human Machine Interface) en ondersteunt verschillende formaten kleurendisplays, van 3,5 inch QVGA (240 × 320) tot 7 inch met 800 × 400 pixels. Connectiviteit wordt geleverd door een multi-protocol draadloze module (met KW41Z, geschikt voor BLE 4.2, Zigbee en Thread), een WiFi module, een NFC-lezer, twee CAN-poorten alsmede Ethernet- en USB-interfaces. Een groot aantal on-board-sensoren voor directe interactie met de gebruiker (inclusief gebarenbesturing) maakt 'smart-connected'-apparaten mogelijk.

Brein	NXP LPC54618 ARM Cortex-M4, 180 MHz, 200 KB SRAM, 512 KB Flash
On-board geheugen	16 Mb Flash
Connectiviteit	Ethernet 10/100, CAN, USB, multi-protocol radio (BLE, Thread en Zigbee), WiFi b/g/n, NFC
Sensors	Temperatuur, 3-assige versnellingsmeter, 3-assige gyroscoop, biocompatibele druk, omgevingslicht.
Uitbreidingen	Arduino shield-connectoren
Voeding	DC-connector, USB



Meer informatie

HANI: www.arrow.com/en/campaigns/arrow-hani-board



USB forever

zonder de universele seriële bus zou de wereld er heel anders uit zien...

Meten aan USB



Om het energieverbruik (of het stroomverbruik) van een USB-apparaat te weten te komen, moeten we ofwel de fabrikant geloven (als hij daar al iets over zegt in de documentatie van het apparaat), of we moeten aan de knutsel — we moeten immers op de een of andere manier bij de aansluitingen van de USB-connector komen om daar de spanning te meten. En voor stroom wordt het nog lastiger, want een conventionele meting met een ampèremeter vereist het onderbreken van de stroomkring. Daartoe moeten we dan de soldeerbout warmstoken, het apparaat in kwestie openen (daar gaat de garantie...) en met losse draadjes aan de slag. Dat is niet echt aantrekkenlijk.

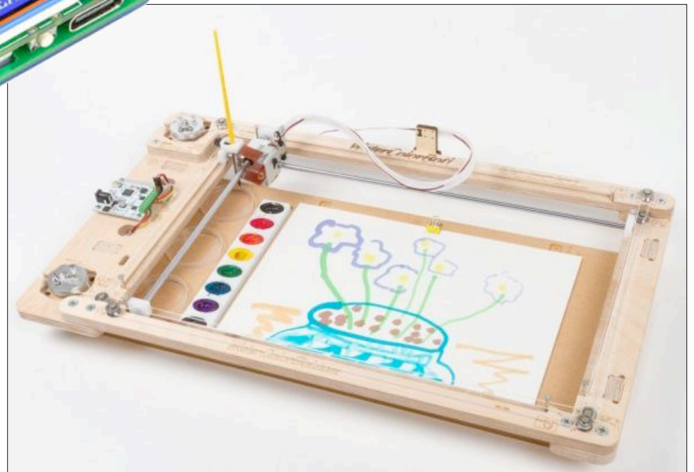
Maar er is een alternatief: een USB-'tussenstekker' die als meetadapter met ingebouwde elektronica dienst doet. De eenvoudigste uitvoeringen van deze meetadapters zijn voor een paar euro te koop; deze beschikken dan over een driecijferig display waarop spanning met een resolutie van 10 mV en stroom met een resolutie van 10 mA wordt weergegeven. En dat laatste is voor veel toepassingen niet nauwkeurig genoeg. Bovendien zijn die goedkope adapters doorgaans niet bestand tegen hogere spanningen dan ongeveer 5 V, zoals bij USB-laders voorkomen (de spanning kan daar bij snelladen tot 20 V oplopen!).

De UM25C brengt daar uitkomst. Deze USB-meetadapter heeft een OLED-kleurendisplay met een respectabele diagonaal van 1,44", waarop plaats is voor vijf cijfers en een oplossend vermogen van 1 mV voor spanningen en 0,1 mA voor stroom.

Het spreekt voor zich dat een luxe meetadapter als deze nog meer functionaliteit in huis heeft. Hij meet niet alleen spanning (4,0...24,0 V) en stroom (0,0...5,0 A), maar ook belastingsweerstand, vermogen, temperatuur, tijd en capaciteit/energie bij laadprocessen (mAh resp. mWh).

Maar het kan nog mooier... Hoewel het display van de UM25C al behoorlijk groot en gestoken scherp is, is een groter display natuurlijk niet verkeerd. En dankzij een Bluetooth-verbinding is dat probleemloos mogelijk: de meetwaarden kunnen op een PC-monitor of smartphone worden bekeken met behulp van speciale (gratis) apps.

Om het geheel af te ronden vermelden we nog dat dit meetwondertje dat in de Elektor-shop op u wacht, ook nog van een datalogger-functie is voorzien... Van harte aanbevolen dus voor iedereen die comfortabel aan USB-aansluitingen wil meten.



Waterverf...

Ter afsluiting van deze aflevering een hebbeding dat elke elektronicus en vooral ook het kind in elke elektronicus zal aanspreken. Een XY-plotter is op zich niets nieuws, en een perfect gereedschap om (technische) tekeningen te maken. Wat echter (voor zover we weten) uniek is aan deze WaterColorBot (versie 2.0) is dat hij — zoals de naam al suggereert — niet alleen met pen en potlood overweg kan, maar ook met waterverf!

Daartoe worden napjes met waterverf in het apparaat naast het tekenblad geplaatst samen met een (petri)schaaltje met water, en een penseel in de houder geklemd. De WaterColorBot doopt het penseel in het water, neemt de juiste kleur verf en volgt daarbij een vectortekening op uw computer (of volgt de cursor in real time terwijl u tekent).

Het aardige daarbij is dat de plotter met alle gewoon in de handel verkrijgbare potloten, pennen, penselen en waterverfnapjes werkt; u bent dus niet gedwongen speciale (en vaak schandalig dure) *consumables* te kopen.

En indachtig de titel van deze aflevering — de WaterColorBot wordt met een gewoon USB-kabeltje op de PC aangesloten. De voor het gebruik vereiste software kan via het internet worden gedownload en ondersteunt Mac-, Windows- en (de meeste) Linux-systemen. ◄

(180471-01)



IN DE STORE

→ UM25C USB-tester
www.elektor.nl/bakeey-um25c



IN DE STORE

→ WaterColorBot
www.elektor.nl/watercolorbot-2-0-pen-plotter

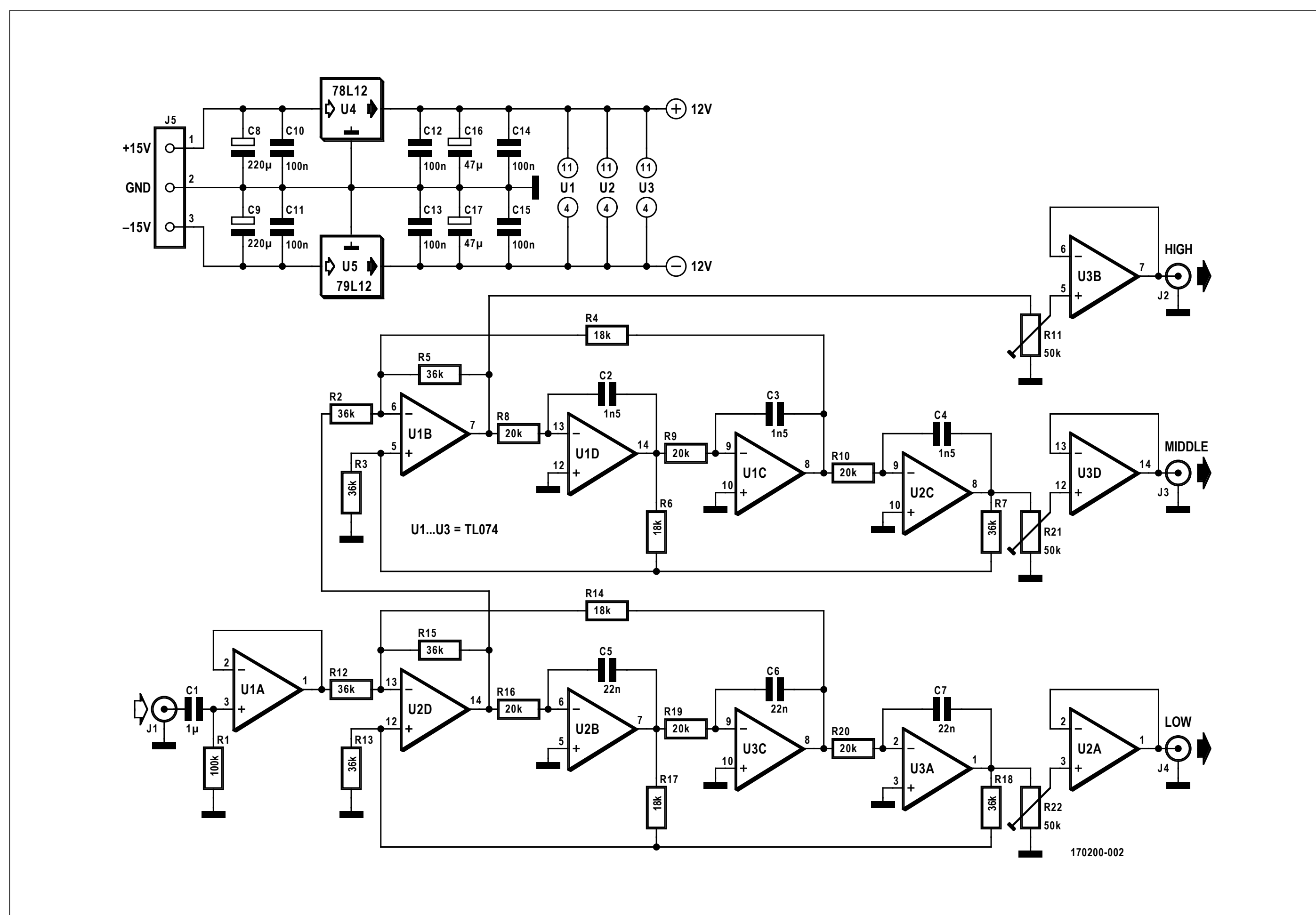
Simpel & goedkoop actief audio-wisselfilter

drieweg, derde- en vierde-orde

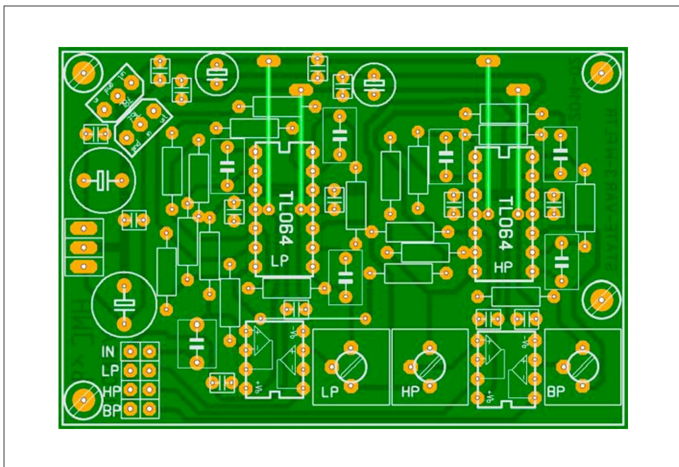
Jac Hettema

In de loop der jaren zijn talloze actieve wisselfilters gepubliceerd, vrijwel zonder uitzondering met veel — en vaak dure — componenten. Het hier gepresenteerde filter is een zeldzame en gunstige uitzondering op deze regel, met relatief weinig en bovendien 'normale' componenten, die niet een precisie hebben tot in de n^de graad en gemaakt zijn van unobtainium™.

Dat zijn enkele van de voordelen van een *state-variable filter*. Het meest toegepaste concept voor een audio-wisselfilter is echter een tweede-orde hoogdoorlaat- (*HighPass*, HP) of laagdoorlaatfilter (*LowPass*, LP), bestaande uit twee weerstanden,



Figuur 1. Basisontwerp van een derde-orde drieweg-wisselfilter.



Figuur 2. Printontwerp voor het derde-orde drieweg wisselfilter (ontwerp van de auteur).

twee condensatoren en een opamp of soortgelijk actief element. Daar hebben we dan met twee RC-netwerken in de schakeling te maken met alle onvermijdelijke onderlinge beïnvloeding.

Een uitstekend alternatief

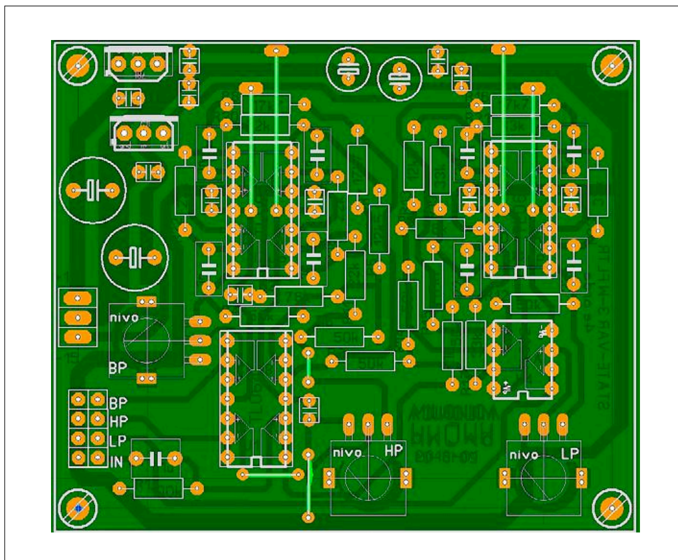
Een state-variable filter bestaat uit een sommerende versterker gevolgd door aantal integratoren. De uitgang van elke integrator wordt met de juiste fase en signaalniveau teruggekoppeld naar de sommerende versterker.

Het aantal integratoren in de schakeling bepaalt de orde van het filter, de karakteristiek wordt bepaald door de mate van terugkoppeling. Zowel Bessel- als Butterworth-karakteristieken zijn mogelijk, de benodigde berekeningen zijn te vinden in op internet beschikbare literatuur [1] [2] en in het boek *The Design of Active Crossovers* van Douglas Self.

In het state-variable zijn alle RC-netwerken van elkaar gescheiden door hun 'eigen' opamp, zodat ze elkaar niet beïnvloeden. Hierdoor kunnen gewone condensatoren (wel van goede kwaliteit) worden gebruikt en worden de correcte RC-tijden bereikt door 1%-weerstanden toe te passen, die tegenwoordig ruimschoots voorhanden zijn (en die veel goedkoper zijn dan 1%-condensatoren).

Daarnaast volstaat slechts één filter om de hoge en de lage frequenties te scheiden, waardoor minder componenten nodig zijn. Overigens is een enkel wisselfilter al voldoende om een tweeweg-geluidssysteem te realiseren; een drieweg-systeem kan met slechts twee filters worden gebouwd.

Figuur 3. Basisontwerp van een vierde-orde drieweg-wisselfilter.



Figuur 4. Printontwerp voor het vierde-orde drieweg-wisselfilter (ontwerp van de auteur).

De kantel- of afsnijfrequenties worden bepaald door de tijdconstanten van de integratoren:

$f = 1 / (2 \pi R C)$

Het is van belang dat de RC-tijden een gelijke lengte hebben; als een condensator teveel afwijking vertoont kan dit worden gecorrigeerd met de 1%-weerstanden.

Schema's & printen graag...

Het schema in **figuur 1** toont een derde-orde actief wisselfilter. De print hiervoor is afgebeeld in **figuur 2**, de met Sprint-Layout 5 gemaakte layout kan worden gedownload van [3].
Uit **figuur 3** blijkt dat een (steilere) vierde-orde variant niet veel verschilt van het derde-orde filter. Hetzelfde geldt voor de printlayout in **figuur 4** ◄.

(170200-04)

Weblinks

[1] Dennis A. Bohn: A Fourth-Order State Variable Filter for Linkwitz-Riley Active Crossover Designs: www.rane.com/pdf/linriley.pdf

[2] Rod Elliott (ESP): State Variable Filters: http://sound.whsites.net/articles/state-variable.htm

[3] Projectpagine bij dit artikel: www.elektormagazine.nl/170200-04



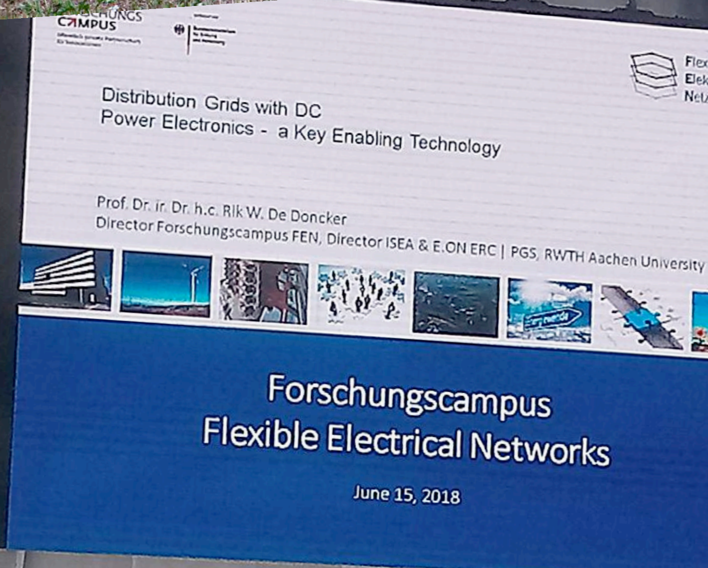
HomeLab Helicopter

Samengesteld door **Clemens Valens** (Elektor Labs)

Wie heeft er nou gelijk?



DC (links) tegen AC (rechts, of was het nu andersom?), wie gaat het winnen?



Van onze vliegende reporter Luc Lemmens

Het is al zo'n 120 jaar geleden dat de zogenaamde "Oorlog der stromen" beslist werd in het voordeel van het wisselspanningssysteem van Tesla en Westinghouse. Edison moest het onderspit delven met zijn gelijkspanningssysteem voor distributie van elektrische energie en sindsdien bestaat ons energienet voor het overgrote deel uit wisselstroomverbindingen. In de tijd van Edison en Tesla was dit een goed idee, maar is dat nog steeds het geval? De meeste (huishoudelijke) elektrische apparatuur werkt intern op gelijkspanning. Bijna elk apparaat heeft daarom een voeding die de wisselspanning van ons lichtnet omzet in gelijkspanning. Bij deze omzetting gaat energie verloren en bovendien zijn het vaak juist deze voedingsmodules die als eerste sneuvelen. Met name de elektrolytische condensatoren die hierin zijn toegepast, hebben een relatief korte levensduur. De voeding van een AC-LED-lamp bijvoorbeeld is vaak snel stuk terwijl de LED's die erin zitten nog heel lang mee zouden kunnen. Een gelijkspanningsnetwerk zou hier zeer zeker voordelen bieden.
Diverse alternatieve energiebronnen, waaronder de zonne-energie die nog steeds in toenemende mate bijdraagt aan onze energiebehoefte, leveren DC en in het gros van deze PV-installaties wordt de geleverde gelijkstroom eerst omgezet

in wisselstroom, zodat ze aan ons lichtnet gekoppeld kunnen worden. Ook die stap zou overgeslagen kunnen worden. Fossiele brandstoffen, die van oudsher een zeer belangrijke rol hebben gespeeld bij onze energieproductie, raken steeds verder uitgeput. Bovendien dwingen strengere milieu-eisen en –doelstellingen (vooral op het gebied van CO_2-uitstoot) ons om meer en meer gebruik te maken van schonere, duurzame energiebronnen. Deze energietransitie brengt heel wat nieuwe uitdagingen en inzichten met zich mee, waarbij op het gebied van de elektriciteitsvoorziening niet alleen de energiebronnen, maar ook het transport en de opslag onderwerp van discussie zijn. Een heroverweging van de toepassing van gelijkspanning in dit geheel ligt voor de hand.

Tijd voor een congres

De Stichting Gelijkspanning Nederland organiseerde op 15 juni 2018 in Nieuwegein het "DC Congres Groot Gelijk 2018", met KIEN Innovatiemeesters, CityTec en TKI Urban Energy als medeorganisatoren en partners van het congres. "Een inspirerende dag over de rol van gelijkspanning in de energietransitie", zoals de organisatie het aankondigde. Deze dag werd bezocht door een gemengd gezelschap van onderzoekers, ontwikkelaars, beleids- en plannenmakers en uitvoerders.

Het ochtendprogramma werd grotendeels gevuld door drie keynote speakers. De middag van het congres was gereserveerd voor kortere deelsessies, waarin verschillende aspecten en ontwikkelingen op het gebied van gelijkspanning aan de orde kwamen.



Keynote sprekers

De eerste spreker, voorzitter van Uneto VNI Doekle Terpstra, belichtte de uitdagingen die de energietransitie met zich meebrengt voor de installateurs. Gelijkspanningssystemen gaan volgens hem daarin zeker een belangrijke rol spelen. Daarnaast waarschuwde hij voor het tekort aan vakmensen, waarbij werd opgemerkt dat met de huidige en toekomstige techniek de informatietechnologie een steeds grotere rol gaat spelen en de vraag naar hoger opgeleid technisch personeel — ook in de installatietechniek — alleen maar zal toenemen. In de huidige situatie kan het Nederlands onderwijs niet in deze vraag voorzien, en al helemaal niet als het op DC-installaties aankomt, alleen al doordat het ontbreekt aan docenten.

Als tweede spreker kwam Ed Buddenbaum van het ministerie van Economische Zaken en Klimaat, secretaris Topsector Energie, aan het bod. Hij besprak de toekomstvisie en het energiebeleid van de Nederlandse overheid, dat er onder meer op gericht is de uitstoot van CO_2 te beperken. Het terugdringen van gebruik van fossiele brandstoffen, onder andere voor de opwekking van elektriciteit, heeft daarbij prioriteit. DC kan daar in zijn visie zeker een rol bij spelen, maar het is geen doel op zich.

De laatste spreker, Dr. ir. Dr. h.c. Rik W. De Doncker (Director of Institute for Power Electronics and Electrical Drives, RWTH Aachen University), voorziet een gouden toekomst voor DC waarbij wisselspanning een ondergeschikte rol gaat spelen in de energievoorziening. Wind- en zonne-energie zijn al in ruime mate voorhanden, maar hun opbrengst is uiteraard weers- en seizoensafhankelijk. In piekuren kan de overproductie worden opgeslagen en die voorraad kan in tijden van lagere productie weer teruggevoerd worden naar het net. Zeker voor energieopslag (bijvoorbeeld in accu's of als waterstof) is DC de meest voor de hand liggende oplossing.

Gemengde gevoelens

Er zijn zeker vele voordelen van gelijkspanning te noemen, maar helaas werd er tijdens het congres voorbijgegaan aan de nadelen die eraan verbonden zijn, bijvoorbeeld corrosie door elektrolyse en de hogere eisen waaraan het schakelmateriaal moet voldoen. En een heel belangrijk nadeel in de huidige situatie: het ontbreekt aan specificaties en normen voor DC-installaties. Het AC-verhaal is in Nederland goed beschreven en dichtgetimmerd in de NEN 1010-normen, voor DC is men nog niet verder dan een eerste versie voor de NPR 9090 (Nederlandse Praktijkrichtlijn), waarbij nog opgemerkt mag worden dat Nederland wat dat betreft mondiaal voorop loopt. Toch heeft het ontbreken van voorschriften ook voordelen; de strakke regelgeving op het wisselspanningsnet is namelijk vaak een beperking voor nieuwe ontwikkelingen.

Heeft gelijkspanning de toekomst?

Of DC de toekomst heeft? Het zal ongetwijfeld een prominentere rol gaan spelen in de elektrische energievoorziening en infrastructuur dan het in de voorgaande decennia heeft gedaan. Maar of het AC zal overvleugelen en zelfs overbodig maken....? Als het aan de organisatoren van het congres ligt uiteraard wel, ook al beseft iedereen dat er nog heel veel moet gebeuren voor het zover is. Eén ding is zeker: de AC/DC-discussie zal er wel wat vriendelijker aan toe gaan dan in de tijd van Edison en Tesla.

HomeLab-hebbedingetje



Net zoals velen proberen om de tientallen afstandsbedieningen die ze in huis hebben te vervangen door één universele, zo zou de serieuze elektronica hobbyist graag zijn collectie programmeertools en -adapters willen vervangen door één universele. Nu is universeel een relatief begrip als het op de werkelijkheid aankomt, maar de MiniPRO TL866A Programmer is toch een stap in de goede richting. Niet alleen kan deze zowel PIC- als AVR-microcontrollers aan, de TL866A ondersteunt ruim 14.000 verschillende devices, van de modernste microcontrollers tot de meest archaïsche (E)PROM's. Daarnaast kan het apparaat de werking van logische IC's (CMOS en TTL), DRAM en SRAM testen. Bovendien wordt de TL866 geleverd met een interessant arsenaal aan accessoires.

www.elektor.nl/tl866a-universal-programmer

ELEKTOR PROJECT

MIXING / MASTERING / PRODUCTION / REMIXES



ELEKTOR-PROJECT

#PML Новый альбом с нашим ремиксом уже в сети!

DJ Elektor Project

Hou je van electronica? Hou je van muziek? Als je op beide vragen "ja" hebt geantwoord, dan hou je vast ook van elektronische muziek. Ook al kan dit syllogisme wat dubieus overkomen, misschien ben je toch geïnteresseerd in de muziek geproduceerd door de Russische artiest, band of DJ (dit is niet helemaal duidelijk) Elektor Project. Als je ook nog een liefhebber bent van dure sportwagens, haal dan wat te drinken en geniet van dit alles door te kijken naar de video Hy3L9f-CBotE, te vinden op een populaire video sharing-website.

https://vk.com/elektor_project_group

Wilt u meedoen? Stuur uw commentaar, suggesties, tips en trucs naar labs@elektor.com

Broodje olifant

Elders in deze rubriek is de "Oorlog der stromen" al aan de orde gekomen. Eén van de offensieven die Thomas Edison voerde tijdens deze oorlog om het gevaar van wisselstroom te laten zien, bestond uit het elektrocuteren van dieren. Het beruchtste voorbeeld dat vaak wordt aangehaald om de wreedheid van Edison aan te tonen, was de elektrocutie van de circusolifant Topsy. Edisons filmproductiebedrijf heeft deze executie destijds op film vastgelegd, maar, als we de historici mogen geloven, was Edison daarbij zelf helemaal niet aanwezig. Sterker nog, op die dag, 4 januari 1903, had Edison de stroomoorlog al zo'n tien jaar verloren. In werkelijkheid was de executie van Topsy de enige manier die haar baasjes Frederick Thompson en Elmer Dundy hadden gevonden om van het beest af te komen. Topsy had een slechte reputatie, waarschijnlijk te danken aan een leven van slechte behandeling, en niemand wilde haar hebben. Om haar te doden werd zo weinig mogelijk aan het toeval overgelaten: gif, wurging met behulp van een stoommachine en elektrocutie om het af te maken…



Moleculen zijn de elektronische bouwstenen van de toekomst

Individuele moleculen kunnen op elkaar aangesloten worden met metalen nanodeeltjes en zo lading door een schakeling transporteren.

https://phys.org/news/2018-07-future-electronics-chemical.html

Hardware-ontwerp met (V)HDL (2)

tijdweergave in binair formaat

Jörg Zollmann (Duitsland)

In het eerste deel van onze VHDL-serie kwam alle basisbegrippen al aan de orde. Aan de hand van een minimaal voorbeeld (een knipper-LED) hebben we kennis gemaakt met de opeenvolgende ontwerpstappen en -tools. In dit tweede deel borduren we hierop voort en concentreren we ons meer op de beschrijving van de hardware met behulp van VHDL.

Tabel 1. Pinning.

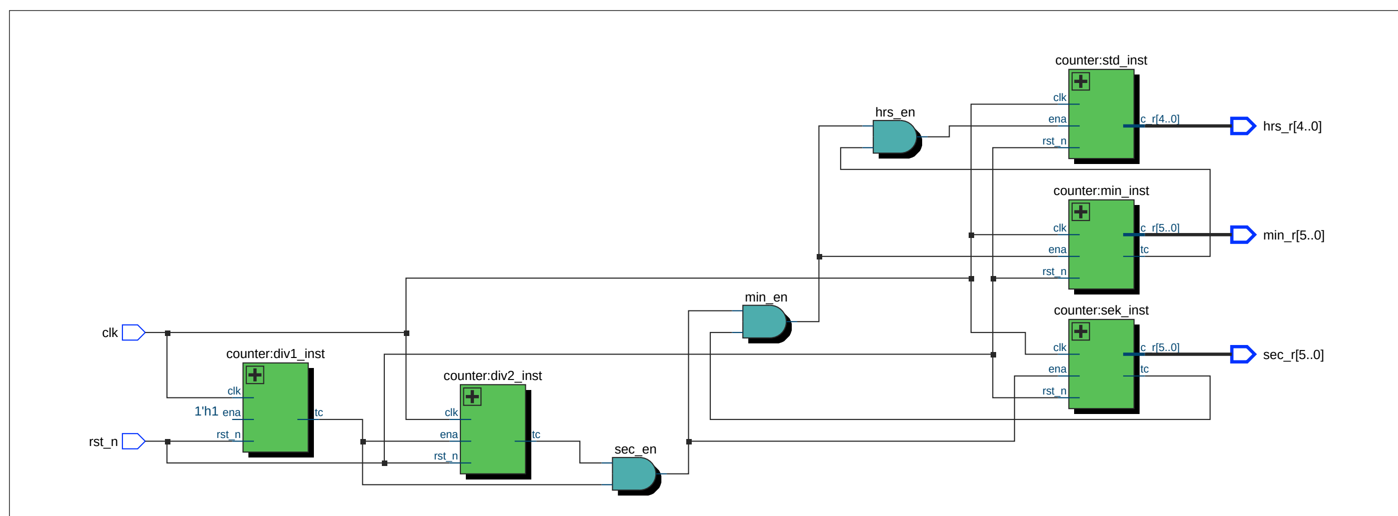
Pin connector K2 (links-boven)	Signaal-naam	CPLD-pin
2	hrs_r[4]	Pin_68
4	hrs_r[3]	Pin_69
6	hrs_r[2]	Pin_70
7	hrs_r[1]	Pin_71
8	hrs_r[0]	Pin_72
9	min_r[5]	Pin_73
10	min_r[4]	Pin_74
11	min_r[3]	Pin_75
12	min_r[2]	Pin_76
13	min_r[2]	Pin_77
14	min_r[0]	Pin_81
15	sec_r[5]	Pin_82
16	sec_r[4]	Pin_83
17	sec_r[3]	Pin_84
18	sec_r[2]	Pin_85
19	sec_r[1]	Pin_86
20	sec_r[0]	Pin_87

In dit artikel bouwen we stap voor stap een digitale klok. Deze geeft de cijfers weer in binair formaat, zodat de klok alleen door hardcore-elektronicaliefhebbers kan worden afgelezen. Voor de uren-aanduiding gebruiken we 5- en voor de minuten en seconden 6-bits waarden.

We beperken ons hier tot een pure simulatie, maar wie dat wil kan natuurlijk de registerwaarden direct via de uitgangen van de CPLD's naar buiten voeren en een rijtje LED's aansluiten (zie **tabel 1**). De hardware bestaat ook in dit deel uit het CPLD-break-out-board (zie het kader 'In de store'). Op de webpagina bij dit artikel staat de projectsoftware als gratis download voor u klaar [2]. De organisatie van het geheel is dezelfde als beschreven bij het project in het eerste deel [3].

Frequentiedeler

Als tijdbasis voor onze klok gebruiken we het 40 MHz-kloksignaal van de CPLD-print. Voor een klok hebben we meerdere tellers nodig. Een secondenteller die elke seconde optelt, een minutenteller en een urenteller. We zouden natuurlijk elk van deze tellers met een eigen kloksignaal kunnen voeden en de tellers de klokpulsen laten tellen. De secondenteller zou dan bijvoorbeeld met een kloksignaal van 1 Hz worden gevoed en de urenteller met 0,278 mHz (1/3600 s^{-1}) enzovoort. Deze aanpak is weliswaar functioneel gezien juist, maar leidt ertoe dat we geen volledig synchrone schakeling hebben. We kunnen beter alle elementen van de schakeling voorzien van hetzelfde kloksignaal en de berekeningen (in dit geval omhoog tellen) alleen dan uitvoeren, als er een klokvrijgave-puls is (*Enable*-impuls (ena)).



Figuur 1. Blokschema van het top-level entity.

Listing 1. Parametriseerbare teller.

```
library ieee;                 -- Use the library ieee
use ieee.std_logic_1164.all; -- std_logic_1164 from the ieee library package has to be applied
use ieee.numeric_std.all;    -- in the numeral package i.e. the typ unsigned is defined

entity counter is  -- Entity declaration
  -- a generic is valid for the entity and the architecture
  -- generics can be overwritten in the generic map during initialization
  generic ( c_width   : natural  := 5;
            c_max_val : natural  := 10;
            reset_val : unsigned );
  port (        -- Port  List
    clk       : in  std_logic;
    rst_n     : in  std_logic;
    ena       : in  boolean;
    c_r       : out unsigned (c_width-1 downto 0);
    tc        : out boolean
    );
end counter; -- End of the entity declaration

architecture rtl of counter is –- Architecture declaration
begin

  tc  <= true when c_r = c_max_val else false; -- every time the counter reaches its
                                               -- maximum value, tc =true
                                               -- if not, then false
  up_counter : process (clk,rst_n) is
  begin
   if (rst_n = ‘0’) then
     c_r <= reset_val; -- after reset he counter starts with the by
                       -- generic defined reset_val
   elsif rising_edge(clk) then
     if (ena) then
        if (c_r = c_max_val) then
          c_r <= (others=>’0’);-- the counter may count until the maximum value and
                               -- thereafter restart from zero
        else
          c_r <= c_r+1;        -- if the counter hasn’t reached its maximum value,
                               -- the value is incremented with 1 every clockcycle (if ena is
                               -- active)
        end if;
     end if;
   end if;
  end process up_counter;
end rtl;
```

Voor onze digitale klok betekent dat, dat de secondeteller elke seconde, de minutenteller elke minuut en de urenteller elk uur een Enable-puls moet krijgen. Daarbij duurt de klokvrijgave-puls slechts één 40 Mhz-klokperiode (25 ns). Maar voor het genereren van een enkele Enable-puls kunnen we eveneens een teller gebruiken, die behalve de actuele tellerwaarde ook een signaal krijgt toegevoerd dat aangeeft of de maximale telwaarde al is bereikt (`tc` = *Terminal Count*). Eén van de basisprincipes van VHDL is het hiërarchische ontwerp van schakelingen. Daarom modelleren wij in ons voorbeeld de digitale klok met behulp van meerdere tellers. **Figuur 1** toont het blokschema van het bovenste hiërarchie-niveau, het zogenaamde *Top-Level-Entity*, dat is gerealiseerd met de RTL-viewer van Quartus. Welke entity of bestand door Quartus als top-level wordt beschouwd, kan worden ingesteld in het *Project Navigator* venster (drop-down van *Files;* klik met rechts op het gewenste bestand; *Set as Top-Level Entity*). In de figuur is te zien dat aan alle tellers hetzelfde kloksignaal (van 40 MHz) wordt toegevoerd, maar dat hun Enable-ingangen worden gevormd uit de logische combinatie van de diverse Terminal-Count-uitgangen.

Tellers

Onze digitale klok bestaat dus uit vijf verschillende tellers. De tellers `div1_inst` en `div2_inst` genereren samen het klokvrijgave-signaal met een frequentie van 1 Hz. Dat gebeurt doordat teller `div1_inst` van 0 tot 9999 telt en dus elke 0,25 ms aangeeft dat de teller de maximale waarde heeft bereikt. Dit signaal wordt als klokvrijgavesignaal voor teller `div2_inst` gebruikt, die dus elke 0,25 ms zijn tellerstand met 1 ophoogt. Na 4000 pulsen (dus na 1 seconde) is de maximale waarde van deze teller bereikt. De andere tellers vormen de eigenlijke digitale klok, omdat deze de seconden,

Listing 2. Testbench voor het testen van de functies log2 en ceil.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.log2;
use ieee.math_real.ceil;

entity testbench is
end entity testbench;

architecture tb of testbench is
begin
    process is
    begin
        report "Wie funktioniert log2 und ceil?";
        for i in 1 to 9 loop
          report  "i= " & to_string(i)&
                  " log2= " & to_string(log2(REAL(i))) & " ceil(log2(i)): "
                  & to_string(INTEGER(ceil(log2(REAL(i)))));
        end loop;
        wait;
    end process;

end architecture tb;
```

minuten en uren tellen en de actuele tellerstand als uitvoer hebben. Omdat de actuele tellerwaarde van de beide `div`-tellers niet van belang is, wordt bij het initialiseren van de tellers de uitgang `c_r` in de *port map* expliciet met open aangemerkt. Dat elke teller een andere maximale waarde kan hebben, wordt in VHDL met behulp van `generic`-parameters opgelost. Deze parameters, die van toepassing zijn op de entity en instantie op het moment van

Listing 3. Package om de tellers te configureren.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

library work;
-- PACKAGE HEADER ============================================================
package count_pkg is

 type t_cnt_config is record
   t_CNT_div1     : natural;
   t_CNT_div2     : natural;
   t_CNT_SEC      : natural;
   t_CNT_MIN      : natural;
   t_CNT_HRS      : natural;
 end record;

-- constants for the synthesis
-- a constant from record-type t_cnt_config
constant p_cnt_config_syn : t_cnt_config := (
  10000,       -- T=10000 * 25 ns  enable Pulse every 0,25 ms
  4000,        -- T=4000 * 0.25 ms  enable Pulse every 1s
  60-1,          -- 60 seconds --> 0-59
  60-1,          -- 60 minutes
  24-1           -- 24 hours
);
-- constants for the simulation
constant p_cnt_config_sim : t_cnt_config := (
  10,
  4,
  60-1,
```

```
  60-1,
  24-1
);
end package count_pkg;
-- PACKAGE BODY ==============================================================
package body count_pkg is
-- The PACKAGE contains only constants, the package body is empty.
-- Here could be function definitions
end package body;
```

de uitwerking, gedragen zich als constanten, maar kunnen worden overschreven vanuit een hoger hiërarchisch niveau (daarover later meer). **Listing 1** toont de *Entity*- en *Architecture*-definitie van de generieke tellers, voorzien van zoveel mogelijk uitleg in de commentaarregels. Rekenkundige bewerkingen (in onze teller het ophogen met 1) zijn in VHDL uitsluitend voor numerieke datatypen gedefinieerd (`integer`, `real`, `signed`, `unsigned`). `unsigned` en `signed` zijn gedefinieerd in de library *numeric_std* en gebruiken intern het datatype `std_logic_vector`. Daarom kiezen we voor de actuele tellerwaarde een signaal van het type `unsigned` en declareren we de beide bibliotheken *numeric_std* en *std_logic_1164*. De teller heeft drie parameters, die de maximale waarde, de resetwaarde (zie hierna) en het aantal benodigde bits (de registerbreedte) bepalen. Het uitgangssignaal `tc` en het klokvrijgavesignaal `ena` zijn van het type `boolean`. De registerbreedte van een teller wordt één hiërarchie-niveau hoger bepaald. Daarvoor worden twee functies uit de *math_real*-bibliotheek gebruikt. Het is natuurlijk ook mogelijk in VHDL eigen functies te schrijven, maar die hebben we pas in een later stadium nodig. Met de functies `log2` en `ceil` kan worden bepaald hoeveel bits resp. registers tenminste nodig zijn om de gewenste maximale waarde te halen. De Testbench uit **Listing 2** demonstreert de toepassing van de beide functies. Deze Testbench kan met de instructie `do sim_tb.do` worden gesimuleerd. Als alternatieve mogelijkheid om kennis te maken en ervaring op te doen met onbekende functies kan de auteur de gratis web-applicatie Eda-Playground aanbevelen [3]. Eda-Playground is een web-based IDE, waarmee op eenvoudige wijze HDL-code kan worden geschreven en gesimuleerd. Daarvoor hoeft u geen enkel programma te installeren, en ook hoeft u geen do/.tcl-scripts te schrijven.

Simulatietijd verkorten

De simulatie van een digitale klok neemt in Modelsim (bij de ingestelde overflows van de tellers) wel enige tijd in beslag. Als we alleen de functie van de tellers

Listing 4. Een deel van het top-level entity.

```
entity watch is
    generic(
      g_cnt_config: t_cnt_config:= p_cnt_config_syn
      -- the default from g_cnt_config are the values for the synthesis
      -- in the Testbench these are overwritten in the generic map with
      -- g_cnt_config => p_cnt_config_sim
    );
    port (        --
     ...
    )
  );
end watch;
architecture rtl of watch is

constant c_CNT_div1         : natural   := g_cnt_config.t_CNT_div1;
-- here the constant becomes a value
-- this happens again during compiling (because it's a constant)
...
begin
...
  div1_inst : entity work.counter
  generic map(
    c_width    => div1_width,
    c_max_val  => c_CNT_div1,  -- the maximum value of the counter depends
                               -- on synthesis or simulation cycle
    reset_val  => to_unsigned(0,div1_width)
  )
  port map (
   ...
  );
...
```

Listing 5. Een tcl-script dat het bestand "time_pkg.vhd" genereert.

```
set filename ../src/time_pkg.vhd

# Actual time
set hour    [clock format [clock seconds] -format "%H"]
set min     [clock format [clock seconds] -format "%M"]
set sec     [clock format [clock seconds] -format "%S"]

# Check file existence
if {[file isfile $filename]} {
    puts "$filename exists"
    puts "--- Datei wird entfernt und neu angelegt"
    file delete $filename
} else {
    puts "create $filename"
}

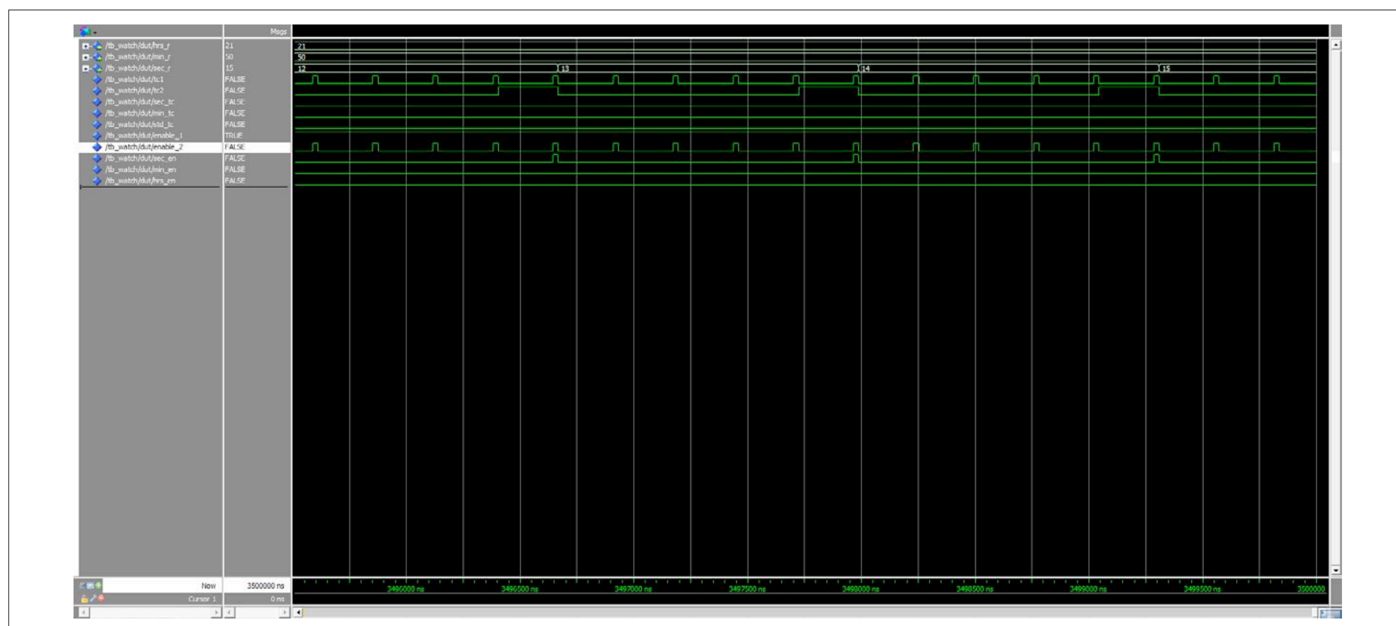
# Open file
if [catch {open "$filename" "w"} f] {
puts stderr $f
} else {
  puts $f "---------------------------------------------"
  puts $f "-- AUTOMATICALLY GENERATED FILE DON'T CHANGE "
  puts $f "-------------------------------------------- "
  puts $f "library IEEE;              "
  puts $f "use IEEE.STD_LOGIC_1164.all;"
  puts $f "use IEEE.NUMERIC_STD.all;   "

  puts $f "library work;"
  puts $f " package time_pkg is"

  puts $f "constant c_hour : unsigned := to_unsigned($hour,5);"
  puts $f "constant c_min  : unsigned := to_unsigned($min,6);"
  puts $f "constant c_sec  : unsigned := to_unsigned($sec,6);"

  puts $f "end package time_pkg;"
  puts $f "package body time_pkg is"
  puts $f "end package body;"
}
# Close file
close $f
```

willen testen, kunnen we de overflows van de tellers tijdelijk op een andere waarde zetten dan in de latere hardware-implementatie. Om niet steeds voor een simulatie de broncode te hoeven aanpassen, maken we gebruik van een VHDL-*Package*. Globale declaraties die op meerdere plaatsen nodig zijn worden in packages samengevat. In packages kunnen functies, constanten en datatypen worden gedeclareerd. **Listing 3** toont het voor de digitale klok gebruikte package *count_pkg.vhd*. Een package bestaat uit een declaratiedeel (*Header*) en een *Body*. Het declaratiedeel vormt de externe interface van een package en is vergelijkbaar met een .h-bestand in de C-programmering. De body van een package bevat doorgaans de definities van de in het declaratiedeel gedeclareerde functies of procedures. In ons package worden geen functies gedeclareerd, zodat dit nu een lege body heeft. In plaats hiervan worden een VHDL-*Record* gedeclareerd en twee constanten gedefinieerd. Met behulp van records kunnen willekeurige typen data worden samengenomen, waarmee abstracte datatypen kunnen worden gebouwd. Records zijn vergelijkbaar met *Structs* in de programmeertaal C. De toegang tot elementen/velden binnen een record gebeurt via de daaraan toegewezen namen. Als een record in een proces wordt gebruikt en aan elementen van dat record waarden moeten worden toegewezen, moet aan álle elementen van dat record een waarde worden toegewezen. In ons geval bestaat het record uit de vijf maximaalwaarden van de tellers en heeft het de naam `t_cnt_config`. De constanten heten `p_cnt_config_syn` en `p_cnt_config_sim`. De namen dienen slechts om onderscheid te kunnen maken tussen diverse functies, ze kunnen vrij worden gekozen (maar het is wel zo gemakkelijk een betekenisvolle en consequente naamgeving te hanteren). De maximaalwaarden van de diverse tellers zijn — zoals eerder aangeduid — door een generic-parameter vastgelegd; de waarden van de generics zijn bepaald door de in het package gedefinieerde constanten. **Listing 4** toont een deel van het top-level-entity. Het top-level-entity bevat ook een generic-parameter van het type `t_cnt_config`. Aan deze parameter is een standaardwaarde toegewezen die de teller-configuratie bevat die nodig is voor de synthese. Bij de initialisatie van de digitale klok in de Testbench wordt de parameterset door een andere overschreven,



Figuur 2. Simulatie.

zodat in de simulatie de klokvrijgave elke milliseconde wordt gegenereerd in plaats van elke seconde.

Instellen van de tijd

Nu rest alleen nog de vraag, hoe een startwaarde kan worden ingesteld voor de uren-, minuten- en secondentellers. Als de tellers straks de werkelijke tijd moeten aangeven en niet alleen als stopwatch fungeren, dan is een variabele startwaarde toch wel het minste. Dat kunnen we bereiken door de Start- en Reset-waarde, net zoals we met de maximaalwaarde hebben gedaan, via een constante uit een separaat package als generic mee te geven. Daardoor kunnen we steeds voordat we een nieuwe compileerslag starten, de actuele tijd centraal in een bestand aanpassen. Deze foutgevoelige actie kunnen we automatiseren door een entry in het bestand *Uhr.gsf* of in een tcl-script. Quartus biedt op verschillende momenten in het compileerproces 'hooks', die de ontwikkelaar in staat stellen scripts uit te voeren. De regel

`set_global_assignment -name PRE_FLOW_SCRIPT_FILE „quartus_sh:time.tcl“`

in het bestand Setting zorgt ervoor, dat aan het begin van een nieuwe doorgang het script met de naam time.tcl wordt uitgevoerd. In **Listing 5** is het tcl-script te zien. Het script genereert een bestand met de naam *time_pkg.vhd* en schrijft in dit bestand de actuele tijd als drie VHDL-constanten (c_hour, `c_min`, `c_sec`) in de vorm van een VHDL-package.

Testbench

De voor de simulatie benodigde Testbench is minimalistisch gehouden en bestaat alleen uit de initialisatie van de digitale klok en nog een proces dat aan het begin het Reset-signaal activeert en aansluitend weer deactiveert.

Op vergelijkbare wijze als in het eerste deel wordt de simulatie gestart, als vanuit het *sim*-pad in het Modelsim-Transcript-venster de opdracht `do sim_watch.tcl` wordt uitgevoerd. **Figuur 2** toont de uur-registers (`hrs_r`, `min_r`, `sec_r`) en de `enable`- en `tc`-signalen. Omdat het volledige ontwerp alleen is te vertalen als het bestand *time_pkg.vhd* beschikbaar is, moet de synthese minstens eenmaal aangeroepen worden voordat de simulatie functioneert.

Vooruitblik

De hier beschreven digitale klok is een voorbeeld van een klein VHDL-project en is vooral bedoeld als inspiratie voor zelfbouw en experimenten. In het volgende deel zullen we onze basiskennis uitbreiden en op engines ingaan. Bovendien zullen we de digitale klok met een betere uitlezing uitrusten, door het schrijven van een display-driver in VHDL.

Wie tot die tijd zijn VHDL-kennis verder wil verdiepen, kan een serie online-tutorials doorwerken [4][5][6].

(180285-B-03)

Weblinks

[1] Hardware-ontwerp met (V)HDL (1): www.elektormagazine.nl/180285-04

[2] Software-download: www.elektormagazine.nl/180285-B-03

[3] Web-based IDE: www.edaplayground.com/

[4] VHDL-Tutorial (Duitstalig): https://de.wikibooks.org/wiki/VHDL-Tutorium

[5] VHDL-Tutorial (Engelstalig): www.nandland.com/vhdl/tutorials/index.html

[6] VHDL-Tutorial (Engelstalig): www.fpga4fun.com/HDLtutorials.html

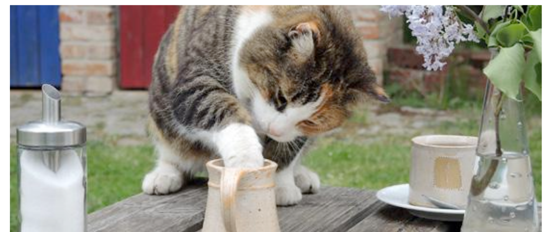
Elektor Labs Pipeline



Meer dan honderd jaar elektronicageschiedenis heeft zijn sporen achtergelaten. Hoewel veel producten uit het verleden al snel in het vergeetboek zijn geraakt, hebben andere het eeuwige leven en houden ons nog altijd bezig...

Van simpel tot ingewikkeld, van meten tot regelen — op Elektor Labs kunt u het vinden, en nog veel meer. Werp eens een blik op deze projecten, er zit vast iets bij dat u kunt gebruiken!

Transistordief
Net als de kat op de foto, die de laatste druppel melk uit het kannetje probeert te halen, is de Joule-dief een (eenvoudige) schakeling waarmee u nagenoeg alle resterende energie uit een bijna lege batterij kunt persen. Vanwege de eenvoud spelen veel mensen (vooral Hollanders) hier graag mee. Volgens Wikipedia is het idee erachter al bijna een eeuw oud. Hier is een wat ingewikkelder variant (ook wel transistordief).



@ Elektor Labs: https://goo.gl/ZDEfM3

Help mee een geheel analoge Theremin te bouwen
Ongeveer honderd jaar geleden uitgevonden, blijft de Theremin een populair elektronisch muziekinstrument. Commercieel verkrijgbare, hoogwaardige exemplaren zijn kostbaar, zodat zelfbouw bijzonder aantrekelijk lijkt. Maar het is nog niet zo eenvoudig een goed reproduceerbare geheel analoge schakeling te ontwerpen. Kunt u misschien helpen?



@ Elektor Labs: https://goo.gl/pq8Dc5

Batterijlader op zonne-energie
Een of meer batterijen met zonlicht opladen — niets nieuws onder de zon; maar een simpele schakeling die u op een regenachtige zondagmiddag in elkaar kunt zetten komt altijd van pas. Hier een voorbeeld. Hij gebruikt een kleine ATtiny13-microcontroller om alles onder controle te houden.



@ Elektor Labs: https://goo.gl/a6eZnM

Red de lavalamp
"Als je mijn lamp koopt, heb je geen geestverruimende middelen meer nodig" — dat was de pakkende slogan waarmee Edward C. Walker de Astro Lamp aanprees die hij in de vroege jaren '60 van de vorige eeuw had uitgevonden. Meer dan 50 jaar later geniet de Lavalamp (zoals hij meestal genoemd wordt vanwege de langzaam opstijgende en zinkende kleurrijke bellen) nog altijd veel belangstelling.

(180461-04)



@ Elektor Labs: https://goo.gl/a6eZnM





Wat wilde u worden toen u jong was?

Dankzij uw ambitie heeft u toegang tot de razendsnel veranderende wereld van engineering, waarin uw ideeën de toekomst vormen. Maar om te doen wat u het beste doet, heeft u tijd nodig om zich te concentreren op wat echt belangrijk is.

Wij helpen klanten al 80 jaar hun doelen te bereiken met ons uitgebreide aanbod procesautomatiserings-, onderhouds- en elektronicaproducten, en op u afgestemde services.

We're here
for the **inspired**



Deskundige technische ondersteuning | Toonaangevende merken | Levering de volgende werkdag

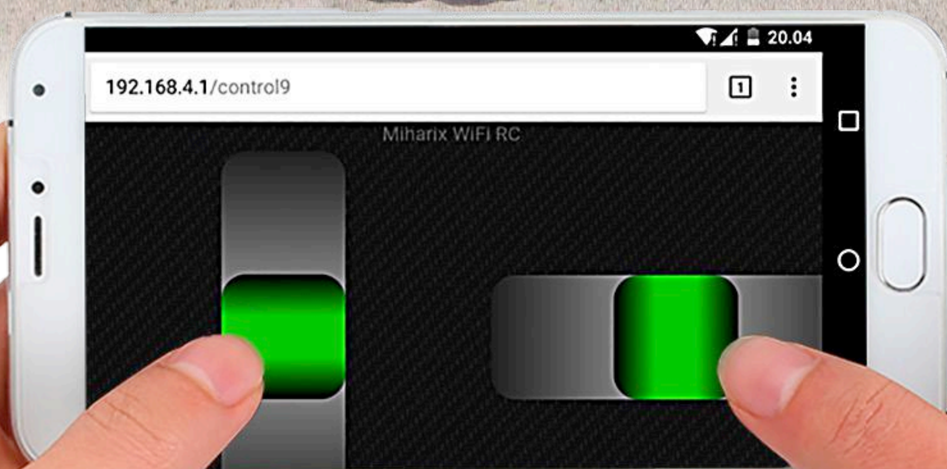
De ESP8266 in de modelbouw

eenvoudige en goedkope afstandsbesturing met een smartphone

Miha Kočar (Slovenië)

De normale draadloze afstandsbedieningen voor modelauto's, -schepen en -vliegtuigen zijn, zelfs in eenvoudige uitvoering, niet altijd zakgeldcompatibel. Maar als we WLAN gebruiken als draadloze verbinding, sparen we om te beginnen de kosten van een zender uit, want elke smartphone is daarvoor geschikt. En de hardware voor de ontvanger is eenvoudig en goedkoop!





De inspiratie voor het ontwikkelen van een WLAN-afstandsbediening voor modellen is ontstaan doordat ik als leraar vaak met dit probleem werd geconfronteerd bij de modelbouwcursus, die ik al enkele jaren op mijn school aan de rand van Ljubljana geef. Meer dan eens ontspon zich een dialoog als deze:

Leerling: "Ik wil ook iets bouwen."
Ik: "Heel goed! Waar denk je aan? We bouwen hier vliegtuigen, boten, zeilboten, auto's..."

Leerling (met grote ogen): "Kun je die ook met een afstandsbediening bouwen?"
Ik: "Natuurlijk. Kijk maar, hier staan verschillende op afstand bestuurbare modellen."

Leerling (met grote, schitterende ogen): "Wow, kan ik dat allemaal hier bouwen?"
Leerling (kijkt ongerust): "Maar is dat niet heel duur?"
Ik (met een vermoeden wat het probleem is): "De materiaalkosten vallen wel mee. We hebben hier op school ook wat bouwmateriaal dat je kunt gebruiken. Het enige wat echt duur is, is de afstandsbediening. Die kost al gauw 100 euro."

Leerling (bedroefd): "...zoveel geld hebben mijn ouders niet."
Ik: "Geen probleem, dan bouw je iets zonder afstandsbediening; die kun je altijd later nog toevoegen."

Leerling (koppig): "Dat is toch stom! Ik wil alleen iets met afstandsbediening!"

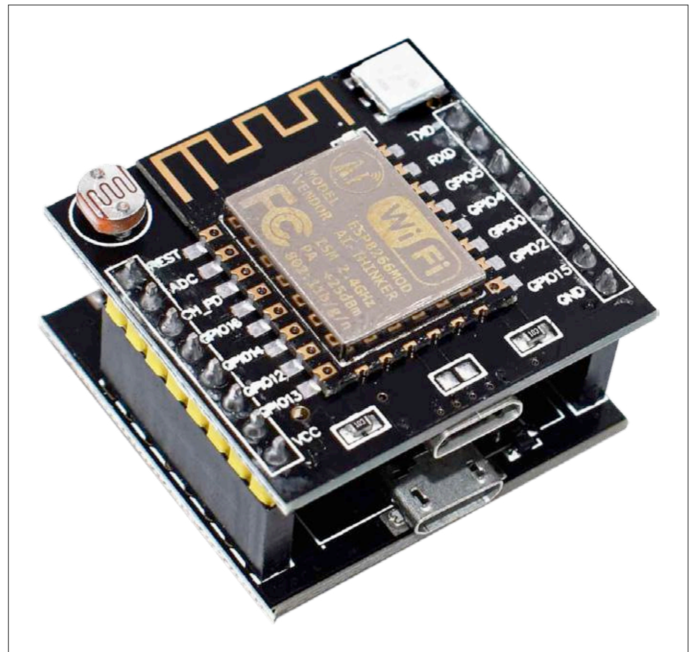
En weer was ik een potentiële modelbouwer kwijtgeraakt. Daar wilde ik iets aan doen...

Achtergrond

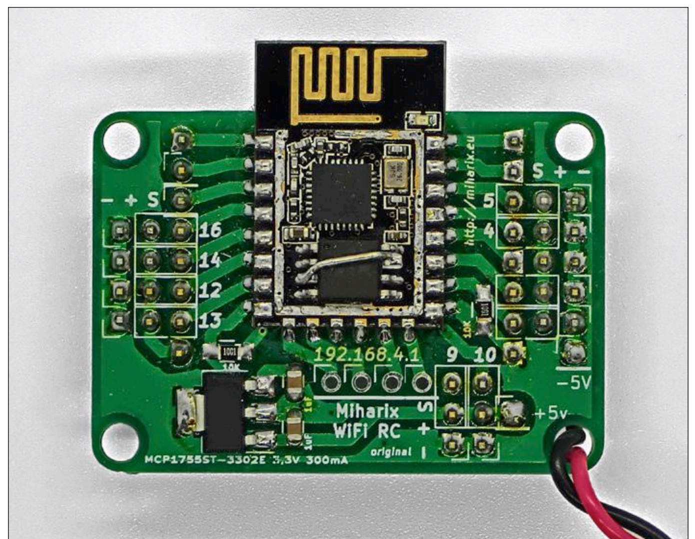
Maar de ironie wil dat alle leerlingen tegenwoordig een smartphone op zak hebben. Daarom stond mij als elektronica-hobbyist een goedkope oplossing voor ogen om modellen met een smartphone te besturen. Helaas vond ik bij mijn zoektocht alleen oplossingen die met een speciale app werken en die natuurlijk niet op elke smartphone draait. En dan was er nog de ESP8266 [1], een inmiddels goed ingeburgerde en vooral goedkope 'WLAN-chip' van de Chinese firma Espressif. Dus begon ik met deze voor WLAN-actuatoren en -sensoren op maat gesneden 32-bit-microcontrollermodule te experimenteren. Tot mijn grote verbazing bleek nog slechts een heel simpele externe schakeling nodig om de in de modelbouw gebruikelijke servo's aan te sturen.
Ik kon bijna niet geloven dat er niet allang een speciaal voor dit doel ontworpen 'China-print' bestond. Maar er waren wel allerlei modules met chip en printantenne en zogenaamde "WiFi Development boards" voor een paar euro te koop bij eBay of AliExpress. Mijn criteria waren duidelijk: zoveel mogelijk smartphone-compatibel, modelbouwcompatibele aansluitingen, gering gewicht, eenvoudig uitbreidbaar, gemakkelijk te programmeren en te bedienen en last but not least goedkoop. Om het gewicht te beperken, zou trouwens een externe programmer de voorkeur verdienen boven een geïntegreerde.

Modules, boards en meer

In eerste instantie koos ik daarom voor de "Witty Cloud" (**figuur 1**), een klein en goedkoop ESP8266-ontwikkelboard. Dit is overal voor weinig geld verkrijgbaar en het bestaat uit twee delen. Het onderste gedeelte is alleen maar een USB-programmerprint. Op de bovenste print bevindt zich een ESP8266-module met al het HF-spul, een RGB-LED, een LDR, een druktoets en een mini-USB-bus. Het is dus eigenlijk een prima uitgangspunt voor kleine modelbouwers, die hun eerste stappen in de wereld van de elektronica willen zetten. Het is geschikt om mee te experimenteren, maar het is niet helemaal optimaal voor modelbouw (voor het aansluiten van servo's).
Voor het ontwikkelen van een eigen, meer geschikte dragerprint (**figuur 2**) koos ik voor een ESP8266-module van het type ESP-14 (**figuur 3**). Deze print is grotendeels compatibel met de gemakkelijker verkrijgbare ESP-12. Mijn kleine print



Figuur 1: De WLAN-ontwikkelkaart Witty Cloud is erg goedkoop en bestaat uit twee printen boven elkaar. Het onderste gedeelte is de programmer.



Figuur 2: Een versie van de door de auteur ontwikkelde print met geopende en voor elf servo's gemodificeerde ESP8266-module.

Figuur 3: De kale ESP8266-module van het type ESP-14 bevat bijna alles wat nodig is voor een afstandsbediening via WLAN, zelfs een printantenne.

bevat naast de module nog een 3,3-V-spanningsregelaar, twee 10-kΩ-weerstanden, een 1-µF-condensator voor het ontkoppelen van de voedingslijn en een (overbrugbare) Schottky-diode als bescherming tegen ompolen. Als we de module vergelijken met de Witty Cloud, valt op dat de pennen TX0, RX0, REST, GPIO0, GND en VCC 1/10" (2,54 mm) naar binnen zijn verplaatst. Om te kunnen programmeren met de programmer van de Witty Cloud is dus een adapter nodig.

Als we naar het schema in **figuur 4** kijken, zien we dat er in principe maximaal elf servo's kunnen worden aangesloten. Maar als we van die mogelijkheid gebruik willen maken, moeten we de module iets aanpassen. Dat is wel mogelijk, maar voor beginners is het vrijwel niet te doen: we moeten daarvoor het deksel van de module openen en binnenin is alles heel erg klein (in figuur 2 is de gescalpeerde en gemodificeerde ESP8266-module te zien). Daarom heb ik dat in de software helemaal weggelaten: alles is ingericht voor maximaal zes servo's. De print heeft dan ook aansluitingen voor zes servo's. Dat

Figuur 4: Het schema van de door de auteur ontwikkelde dragerprint (figuur 2) is heel eenvoudig, omdat niveau-omzetters voor het aansturen van servo's niet nodig zijn.

is ruim voldoende voor het op afstand bedienen van een simpele modelauto of een eenvoudige robot op 'leerling-niveau'. De typische Elektor-lezer is over het algemeen wat ouder en meer ervaren, dus wie het aandurft kan in het tekstkader lezen hoe de volledige implementatie met 11 servo's in zijn werk gaat (op eigen risico...).
Modelbouwers weten, dat we servo's, ontvangers e.d. kunnen voeden met 5 V of soms zelfs met 6 V. Maar de ESP8266 is een 3,3V-chip, die ook met 3,3 V wordt gevoed. Je zou dan vermoeden dat er een soort niveau-omzetter nodig is om de servo's aan te sturen. Maar op veel websites is te lezen, dat het ook zonder gaat! Ik kon zelf in mijn verzameling geen enkele servo vinden, die niet was aan te sturen met maar 3,3 V. En daarom is deze schakeling ook zo extreem simpel.

Programmeren

Eerst een tip: als het niet lukt een nieuw Witty Cloud-board te programmeren, kan dat heel goed liggen aan een slechte USB-kabel. Probeer in dat geval dus eens een andere kabel.
Als programmeeromgeving gebruiken we de vertrouwde Arduino-IDE. Ik gebruik het liefste een portable installatie. Maak daarvoor gewoon een directory met de naam "portable" aan in de Arduino-installatiedirectory [2]. Ga na de installatie naar *Bestand Voorkeuren* en voer bij *Additionele Bordenbeheerder URLs* de onder [3] weergegeven URL in. Zoek dan onder *Schets Bibliotheek gebruiken Bibliotheken beheren...* naar "ESP8266" en installeer *ESP8266, ESP8266HTTPClient, DNSServer, ESP8266WebServer, ESP8266WiFi* en *Servo(esp8266)*. Dan is alles klaar voor het programmeren van eenvoudig interfaces voor langzame of stationaire modelbouw-objecten. Waarom alleen 'langzame'? Omdat de besturing werkt door het aanroepen van een aantal links. Dat kan soms wel even duren.

De structuur van de software voor deze eenvoudige besturing is als volgt: aan het begin van de sourcecode vinden we #include-commando's voor de bibliotheken *ESP8266WiFi.h, WiFiClient.h, ESP8266WebServer.h* en *Servo.h*. De eerste twee library's zorgen voor de WLAN-communicatie, *ESP8266WebServer.h* ondersteunt een WLAN-hotspot en *Servo.h* dient voor het genereren van de PWM-signalen voor de servo's. Daaronder kunnen we het wachtwoord en de SSID (naam van het WLAN) specificeren als "const char *". Het wachtwoord moet langer dan 8 tekens zijn. Het compileren en uploaden zal ook wel werken met een korter wachtwoord, maar dan kan er toch geen verbinding worden gemaakt.
Voor elke aan te sturen servo is een globale variabele nodig. Met het commando `ESP8266WebServer server(80)` starten we een HTTP-server van poort 80 van de ESP8266.
Daarna komen servo-functies, waarmee we de servo's in een bepaalde stand kunnen zetten: `servo_x.write(Grad)` of `servo_x.writeMicroseconds(uS)`. Aan het einde van deze functies moet nog het commando `server.send(200, "text/html",respondX)` staan. In de string `respondX` moet de HTML-code staan, die de content genereert die na het uitvoeren van de eigenlijke servo-functie op het beeldscherm van het mobiele apparaat wordt weergegeven in de browser. De beginpositie van de servo's wordt ingesteld in de setup-functie met `servo_x.attach(pin)` en `servo_x.write(Grad)`.
Dan wordt de WLAN-hotspot geactiveerd met de commando's `WiFi.SoftAP(ssid, password)` en `IPAddress myIP = WiFi.SoftAPIP()`.

11 servo's

Voor de onbevreesden onder ons hier de ingrepen die nodig zijn om alle elf mogelijke servo-uitgangen te gebruiken:

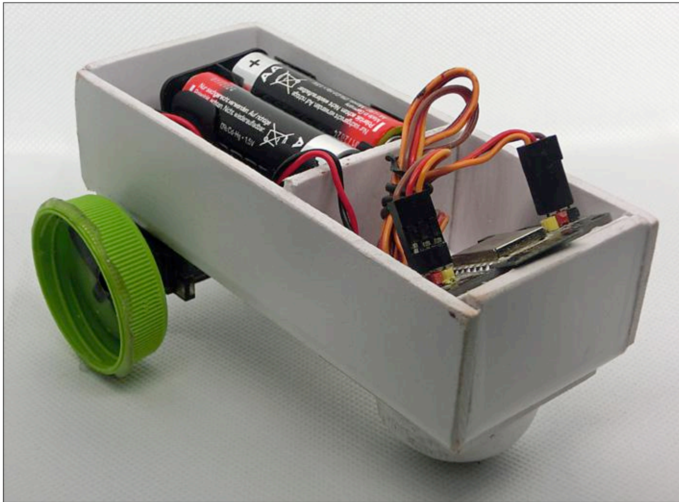
- Verwarm het metalen deksel van de ESP8266-module met een hetelucht-soldeerpistool.
- Probeer dan, het deksel eraf te nemen (met een schroevendraaier, maar voorzichtig!).
- Soldeer de pennen 3 en 7 van de flash-chip los en buig ze omhoog.
- Verbind deze pennen (3 en 7) nu met pen 8 (VCC).

Klaar. In figuur 2 zien we hoe dat eruit ziet. Er zijn op deze print alleen headers voor de massa- en voedingslijnen van zes servo's aanwezig, maar de vijf besturingslijnen voor de andere servo's zijn wel op de aansluitpinnen beschikbaar.

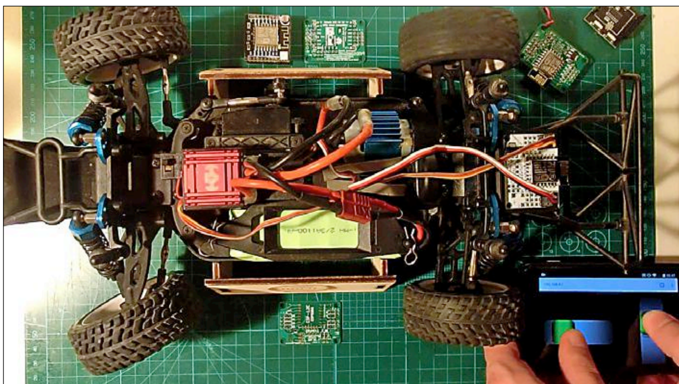
Nu hoeven we alleen nog voor elke gewenste servopositie een link te maken en die met de juiste functie te verbinden:
Voorbeeld 1: `server.on("/",Root)` zal de functie `Root()` uitvoeren via het adres *http://192.168.4.1/*.
Voorbeeld 2: `server.on("/l5", L50)` zal de functie `L50()` uitvoeren via het adres *http://192.168.4.1/*.
Na dat alles komt nog het commando `server.begin()` voor het starten van de HTTP-server.
Tenslotte staat in de functie `loop()` nog het commando `server.handleClient()`, zodat elk HTTP-request van de client (smartphone) zo snel mogelijk wordt uitgevoerd.

Interfaces

We kunnen met deze simpele interface een mobiel apparaat bereiken door daarop eerst een verbinding met de WLAN-hotspot van de ESP8266 op te bouwen en dan met een willekeurige webbrowser naar het bijbehorende IP-adres te gaan, bijvoorbeeld "http://192.168.4.1."
Naast de beschreven, nogal rudimentaire, software-interface bestaat er nog een meer geavanceerde oplossing die te vinden is op GitHub [4]. Die software werkt niet met eenvoudige links, maar gebruikt *WebSockets* voor de besturing van de servo's. Het geheel werkt met een webpagina met allerlei besturingselementen en bestanden van de typen *.html*, *.css*, *.jpg* en *.js*. Met WebSockets wordt de tijd tussen het geven van het commando en het uitvoeren daarvan door de servo enorm verkort, waardoor de besturing ook bruikbaar wordt voor iets snellere modellen zoals auto's en boten. Deze uitgebreide interface werkt via het standaard-adres *http://192.168.4.1*. Maar we kunnen via *http://192.168.4.1/setup* (en */admin*, */root*, */config*, */edit*, */administrator*, */uredi* en *astavi*) van alles en nog wat instellen. We kunnen daarmee de SSID, het wachtwoord, de keuze tussen zelf een hotspot opzetten of koppelen aan een bestaand WLAN en een "fail safe timeout" instellen. Voor elke servo kunnen we individueel de parameters *Trim center*, *Max/Min-uitslag*, *FailSave Position*, *Expo*, *snelheid*, *omkeren van de draairichting* en (tot nu toe) negen verschillende *interface-modi* instellen. Daarnaast kunnen alle instellingen worden teruggezet naar de standaardwaarden via jumpers of een reset. Dat helpt bij fouten met het invoeren van SSID/Password-en bij het koppelen aan een bestaand WLAN. Voor elke chip kan een unieke SSID worden vastgelegd.



Figuur 5: Een heel eenvoudige robot, gebouwd door een leerling van de auteur.



Figuur 6: Ook deze ingewikkelder modelauto is voorzien van een WLAN-afstandsbediening en gebouwd door een leerling.

De software is zo opgebouwd, dat hij ook heel gemakkelijk kan worden gebruikt voor het besturen van andere dingen. Om niet alles twee keer te hoeven beschrijven gaan we alleen in op de bijzonderheden:

- Fail safe timeout: als de chip gedurende x seconden geen nieuwe commando's krijgt, is waarschijnlijk de verbinding verbroken. De servo's gaan dan naar de vastgelegde FailSafe-positie.
- Expo: een veelgebruikte functie voor het omzetten van de lineaire bewegingen van een stuurknuppel naar niet-lineaire servobewegingen. Zo kan de nauwkeurigheid van de stuurknuppel in het midden worden vergroot.
- Koppelen aan een bestaand WLAN: we kunnen een ESP8266 ook koppelen aan een bestaand WLAN. Het IP-adres is dan het adres van de lokale DHCP-server.
- Mode: rechtstreekse aanroep van diverse interfaces met "/controlX" achter het adres, waarbij "X" het interface-nummer is van een bepaalde functie. Voorbeeld: *http://192.168.4.1/control3*.

Bestandsfuncties

Om bestanden met de extensie *.html*, *.css*, *.jpg* of *.js* te uploaden naar de chip, is bij Arduino de plugin SPIFFS (Serial Peripheral Interface Flash File System) nodig. Daarmee kan de chip het flashgeheugen gebruiken als massageheugen om bestanden en directory's te maken, te lezen en te wissen. De plugin wordt geïnstalleerd door hem gewoon uit te pakken in */Arduino/tools/*. Maak in de Arduino-projectdirectory een nieuwe directory "data" aan. Na opnieuw opstarten vinden we dan onder tools een nieuwe entry met de naam *ESP8266 Sketch Data Upload*. Daarmee kunnen we de complete inhoud van de directory "data" naar het flashgeheugen van de ESP8266 kopiëren.
Let op: tijdens het uploaden moet de "Serial Monitor" afgesloten zijn. Het uploaden kan trouwens enkele minuten duren.
Er zitten bij deze softwareversie nog drie extra #includes in de Arduino-code:

1. *FS.h*: nodig voor het werken met de SPIFFS.
2. *WebSocketsServer.h*: voor de socket-server op de Arduino.
3. *ArduinoJson.h*: vergemakkelijkt het werken met het json-bestand.

In het json-bestand komt de specifieke configuratie te staan. De configuratie is direct en indirect toegankelijk via de browser (*http://192.168.4.1/config.json*). Dat vergemakkelijkt het implementeren van een standaard basisinstelling, het opslaan van een tijdelijke configuratie (tot de volgende reset of opnieuw starten) en een overzichtelijk paginaontwerp.
Maar hoe moeten we met SPIFFS de inhoud van het geheugen uitlezen? Voeg na de #includes in de Config-functie nog de regel `SPIFFS.begin()` toe. Daarna kunnen we na het commando `file MijnBestand = SPIFFS.open("/Data.txt", "r")` de inhoud van het bestand *data.txt* inlezen met `MijnBestand.Read()`, net zoals we een normale seriële ingang zouden uitlezen. We blijven lezen, totdat `MijnBestand.available()` de waarde `true` teruggeeft. Na het inlezen moeten we het bestand nog afsluiten met `MijnBestand.close()`.
Schrijven gaat op dezelfde manier: in dit geval zijn de commando's `file MijnBestand = SPIFFS.open("/data.txt", "w")` en `MijnBestand.write()`. Aan het einde sluiten we het bestand weer af. Het uitlezen van het json-bestand via SPIFFS gaat dan als volgt: met `file configFile =SPIFFS.open("/config.json", "r")` openen we het bestand om te lezen en met `size_t size = configFile.size()` bepalen we de grootte. Nu moeten we met `std::unique_ptr buf(new char[size])` een tijdelijke buffer aanmaken en die met `configFile.readBytes(buf.get(), size)` vullen met data. De buffer voor de JSON-parser maken we aan met `StaticJsonBuffer<1024> jsonBuffer` en met `JsonObject& json = jsonBuffer.parseObject(buf.get())` parsen we de inhoud. Nu kunnen we de inhoud van het json-bestand gemakkelijk uitlezen zoals uit een array (voorbeeld: `RCmode = json["Mode"]`). Data in `json` opslaan werkt op een vergelijkbare manier, natuurlijk zonder parser, maar aan het einde komt nog het commando `json.prettyPrintTo(configFile)`.
We kunnen de inhoud van een bestand met streamFile naar de browser sturen. Dat doen we door het bestand te openen en het te verzenden met `size_t sent = server.streamFile(file, contentType)`, waarbij de string contentType is gevuld met het juiste MIME-type. Voor .html-Bestanden is dat dus *"text/html"*.

Websockets

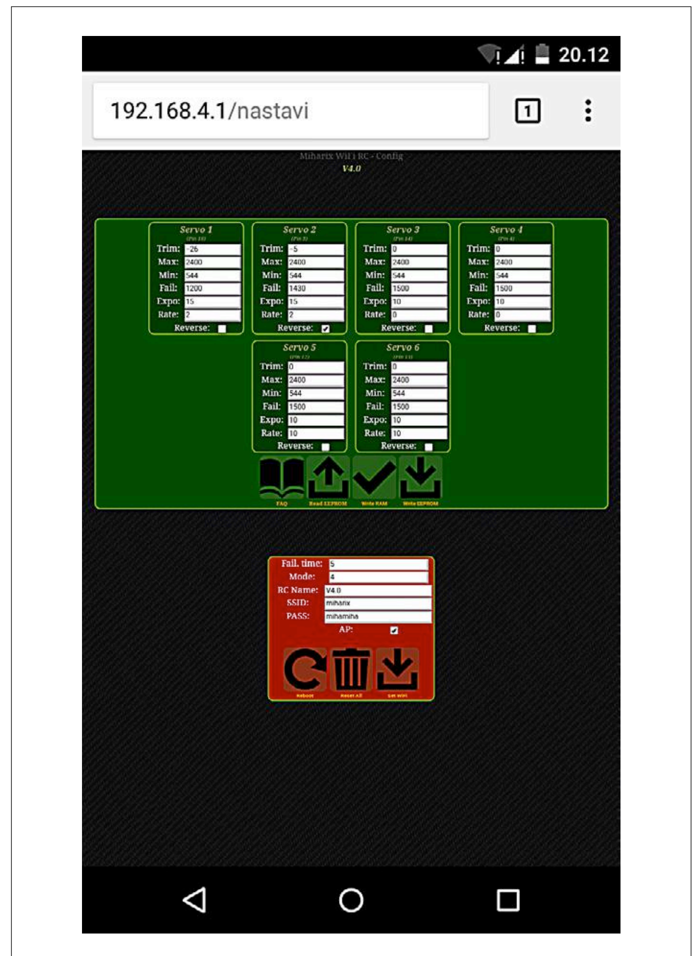
Websockets werken als volgt: Na de #include geven we met `WebSocketsServer webSocket = WebSocketsServer(81)` aan,

dat de websocket-server naar poort 81 moet luisteren. Daarna moeten we in `setup()` een `webSocket.begin()` opnemen om de server te starten en met `webSocket.onEvent(webSocketEvent)` de functie `webSocketEvent` koppelen aan de WebSocket-events. Na `void webSocketEvent(uint8_t num, WStype_t type, uint8_t * payload, size_t length)` kunnen we dan via `IPAddress ip = webSocket.remoteIP(num)` het IP-adres van de client aangeven en met `webSocket.sendTXT(num, "Hallo!")` data naar de client zenden en met een array zoals payload[0] ook uitlezen. De client is in dit geval de webbrowser van het mobiele apparaat, dat de JavaScript-code uitvoert.
In het .js-script staat aan het begin van de code `var connectie = new WebSocket('ws://'+location.hostname+':81/', ['arduino'])` voor het opbouwen van de verbinding. Als we een datapakket willen verzenden, gaat dat met `connection.send('Hallo server!')`. Inlezen van de ontvangen data gaat met `connection.onmessage = function (e) { console.log('Server: ', e.data) }`, waarbij het resultaat in "console.log" alleen zichtbaar is in de ontwikkelaarsmodus van de webbrowser. (In Firefox activeren we die bijvoorbeeld met *ctrl+shift+K*.)
Enkel van de geïmplementeerde interfaces bevatten code, die stuurknuppels realiseert. Deze code is gemaakt met behulp van "nippleJS" en "yoannmoinet".
De sourcecode voor dit project is onder [4] als download beschikbaar.

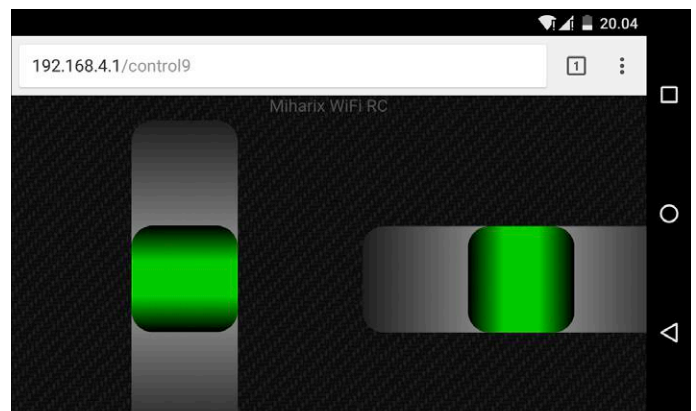
Conclusie

Enkele voorbeelden van wat de leerlingen van de auteur met deze 'Smartphone-afstandsbediening' hebben gemaakt: in **figuur 5** zien we een super-simpele robot, die uit niet veel meer dan twee aangedreven wielen plus batterijen en elektronica bestaat, en in **figuur 6** een wat ingewikkelder modelauto. Let er wel op, dat zo'n WLAN-afstandsbediening maar een bereik van niet meer dan 80 tot 90 m heeft. Het is een heel goedkope oplossing, maar een smartphone kan natuurlijk niet de functie van een stuurknuppel vervullen. In **figuur 7** zien we de configuratiepagina op een smartphone. **Figuur 8** is een screenshot van het besturingsscherm met een horizontale en een verticale slider (dit is één voorbeeld, er zijn intussen negen varianten geïmplementeerd). Bij de vormgeving van zulke interfaces kunnen we onze fantasie de vrije loop laten.
Mocht de auteur ooit besluiten om de software te verhuizen naar GitLab, dan leest u dat op zijn webpagina [5]. Daar vindt u, naast verdere informatie (ook in het Engels), layout-bestanden van de print, software en video's. In de video's wordt weliswaar Sloveens gesproken, maar ze laten duidelijk de mogelijkheden zien. Wie nog meer achtergrondinformatie wil over de hardware en de software, kan via Google veel informatie vinden met de zoektermen "ESP8266" of "Witty Cloud".

(180281-04)



Figuur 7: Screenshot van de configuratiepagina van de WLAN-afstandsbediening met ESP8266.



Figuur 8: Screenshot van één van de tot nu toe negen geïmplementeerde interfaces van de WLAN-afstandsbediening met ESP8266. We zien een verticale en een horizontale slider.

Weblinks

[1] ESP8266: https://bit.ly/2zIKRd7

[2] Portable installatie: www.arduino.cc/en/Guide/PortableIDE

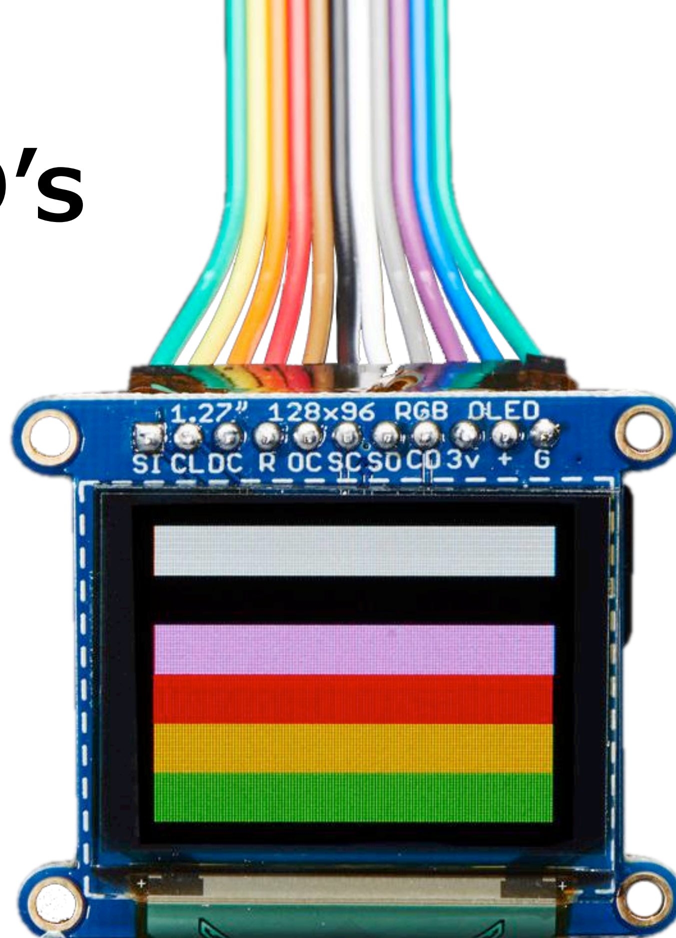
[3] Board-manager: http://arduino.esp8266.com/stable/package_esp8266com_index.json

[4] GitHub: https://github.com/miharix/miharix-wifi-rc

[5] Website van de auteur: http://wifi-rc.miharix.eu

Kleuren-OLED’s

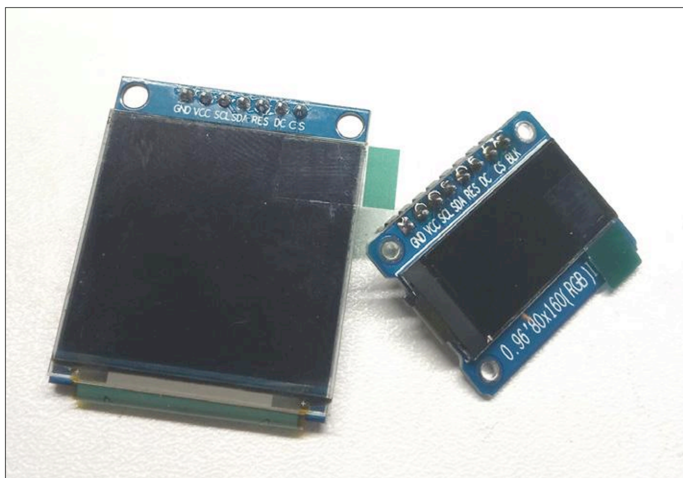
aansturen met STM32



Tam Hanna (Slowakije)

Een kleuren-OLED kan een project veel aantrekkelijker maken; de geïntegreerde besturingschips zijn toegankelijk via SPI. Goedkope 32-bit controllers, geavanceerde bibliotheken en eenvoudig te gebruiken ontwikkelomgevingen maken de aansturing eenvoudig, zoals we in dit artikel laten zien.

Als een klassieke 8-bit Arduino informatie moet weergeven, ligt een monochrome OLED voor de hand. Die heeft maar weinig geheugen nodig en kan eenvoudig worden bestuurd met kleinere processoren.
Maar buiten de Arduino staat de wereld niet stil. 32-bit controllers, vooral van ST, zijn enorm in prijs gedaald. Degenen die graag in high-level talen programmeren, kunnen STM32-chips vinden die wat betreft prijs vergelijkbaar zijn met grote PIC’s. Bijna toevallig zorgde de verdere ontwikkeling van de hardware ervoor dat organische displays goedkoper werden. Solomon Systech heeft de SSD1306 OLED-controller een broertje gegeven in de vorm van de SSD1351 — een nieuwe, verbeterde versie die kleurenschermen tot 128 x 128 pixels aan kan.



Figuur 1. Goedkoop verkrijgbaar bij AliExpress: organische kleurendisplays (foto: tam.hanna op Instagram)

Chinese postorderbedrijven reageren op deze situatie door organische displays aan te bieden in de vorm van de evaluatieboards van **figuur 1**. Het display van 128 x 128 pixels kost minder dan 20 €; zorg ervoor dat u een display met een SSD1351-controller te pakken krijgt.

Ontwikkeling!

In technisch opzicht is de SSD1351 — zoals weergegeven in **figuur 2** — vergelijkbaar met zijn voorganger. Hij gebruikt een SPI-interface voor de communicatie.
Interessanter is de vraag naar de ontwikkelomgeving. ST kocht Atollic een paar maanden geleden op en beschouwt de — ooit schandalig dure — geïntegreerde ontwikkelomgeving “TrueSTUDIO” inmiddels als *best practice*. Als codegenerator moet CUBE worden gebruikt. ST wees de auteur er enige tijd geleden op dat in plaats van de CUBE-plugin een zelfstandige versie gebruikt kan worden (TrueSTUDIO plus een afzonderlijk geïnstalleerde stand-alone CUBE) die stabieler blijkt te zijn. Gebruik alleen de nieuwste versies van de producten. Dit project is gedaan onder Ubuntu 14.04 — zorg ervoor dat u geen verouderde versies gebruikt en installeer onmiddellijk alle updates. Als evaluatiebord kan men elke Nucleo gebruiken. In de volgende stappen willen we een STM32L476 gebruiken — een even krachtig board dat goedkoop en uit voorraad verkrijgbaar is in de Elektor-shop (zie het kader ‘In de Elektor Shop’).
Projecten starten in CUBE — onder Ubuntu kan het programma worden gestart met de volgende opdracht (het gebruik van sudo is alleen vereist voor updates — het wordt aanbevolen om CUBE normaal uit te voeren als een gewone gebruiker):

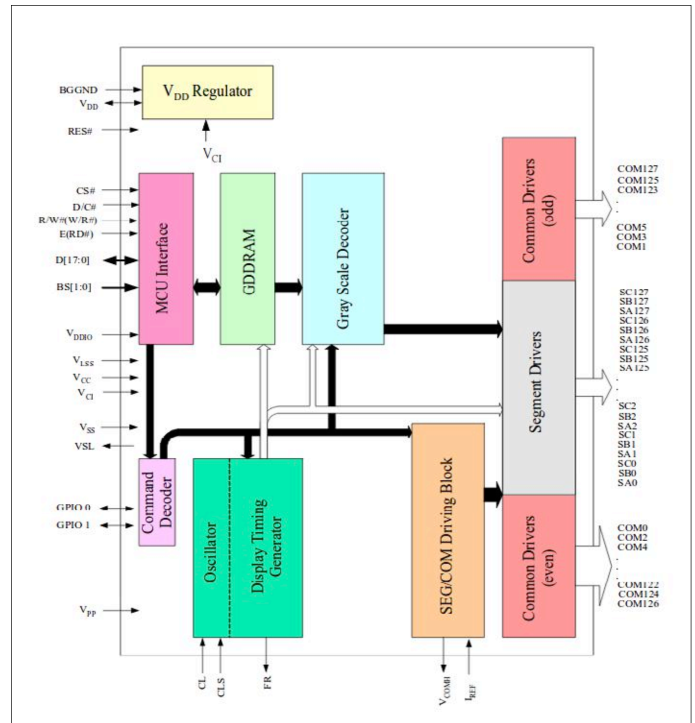
tamhan@TAMHAN14:~/STM32CubeMX$ sudo ./STM32CubeMX

Klik in de volgende stap op *New Project* en ga naar de *Board*

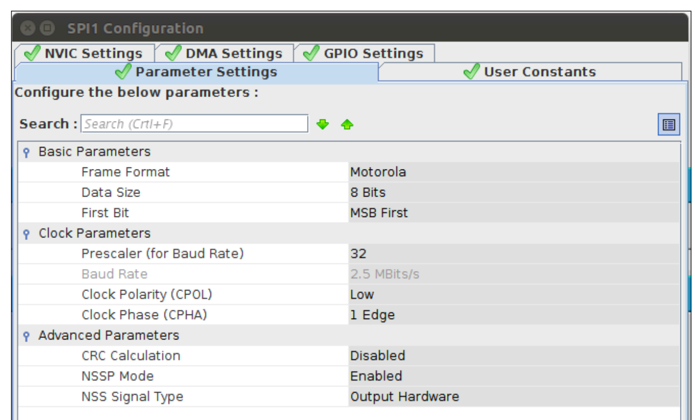
Selector. Zoek naar de string “NUCLEO-L476RG” — als u de naam van de CPU invoert, krijgt u niet het juiste board. De initialisatie van alle randapparaten met standaardwaarden wordt bevestigd, waarna u onmiddellijk een uitvoerbare configuratie krijgt.
We hebben nu eerst een SPI-interface nodig die verantwoordelijk is voor het uitvoeren van de gegevens. Aan de linkerkant is een groep randapparatuur in de categorie *Pinout*, die kan worden geëxpandeerd — vergelijkbaar met een normale ‘boom’ in een Windows-programma.
STM heeft op het Nucleo-board gezorgd voor een groep headers waarmee een pin-out mogelijk is die doet denken aan de Arduino. Vervelend is wel dat de “hoofd-SPI-poort” geblokkeerd is. Eerst moeten we pin PB3 aanklikken en op *Reset State* klikken om de configuratie te wissen. ST reserveert de pen voor JTAG wat niet vereist is bij het werken met de ST-Link.
In de Auth-structuur verdwijnt het X-symbool voor de regel SPI1. Vouw het uit en selecteer in de ComboBox *Mode* de optie *Transmit Only Master.* Klik in het veld *Hardware NSS Signal* op *Hardware NSS Output Signal* — we willen dat de SPI-engine het CS-signaal genereert.
Controleer vervolgens of de pinnen correct zijn toegewezen — STM-randapparatuur is zeer flexibel en de toewijzing kan fout zijn. Dit is vervelend — vooral omdat we de signalen niet kunnen ontvangen bij een foutieve toewijzing aan die pinnen, zoals de Arduino-standaard voorschrijft. PA5 krijgt de groene LED toegewezen; wilt u de Arduino-pinout gebruiken dan moet deze opnieuw gedefinieerd worden.
In de volgende stap hebben we nog twee klassieke GPIO-pennen nodig: ten eerste Reset om de controller als geheel te besturen; ten tweede DC om de SPI-transceiver aan te sluiten op de framebuffer of het parametergeheugen.
De Arduino-pin D8 komt in CUBE overeen met pin PA9: klik erop en selecteer de optie *GPIO_Output*. D7 luistert naar de naam PA8: klik ook hier weer met de rechtermuisknop en selecteer vervolgens *GPIO_Output*. In de volgende stap klikken we op de tab *Configuration*, en daar op de knop *GPIO-Pin* in het gedeelte *Systeem* om het tabblad GPIO te activeren. In het veld *User Label* kunnen we ‘vriendelijke’ namen aan de pinnen toewijzen waaronder ze later in de code toegankelijk zijn.
ST configureert de SPI-engine van de chip standaard agressief. Klik op het tabblad *Configuration* op *SPI1* en stel de parameters in zoals weergegeven in **figuur 3**.
Om de code aan te maken, klikt u op *Project Generate Code*. Als projectnaam kiezen we “ElektorOLED”, in de categorie *Project Location* moet u de werkdirectory van de Atollic-installatie selecteren. Op het werkstation van de auteur is dat */home/tamhan/Atollic/TrueSTUDIO/STM32_workspace_9.0/*. In het veld *Toolchain/IDE* moet u *TrueSTUDIO* selecteren. Nadat u op OK hebt geklikt, downloadt het programma ongeveer 700 MB aan broncode en wordt het project gecreëerd.

Daar gaan we!

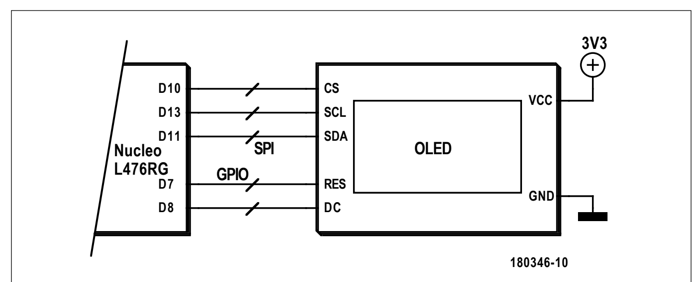
Onze volgende stap is het bouwen van de schakeling van **figuur 4**. Zoals altijd bij Elektor kan de code gratis worden gedownload van de website bij dit artikel [1].
Start nu Atollic en klik op *File Open projects from File system.* Navigeer naar het project en laad het zoals gebruikelijk. Met het .ioc-bestand kunt u het project “herladen” in CUBE — een taak waar we op dit moment echter niet in geïnteresseerd zijn. Voordat u met de eigenlijke programmeerwerkzaamheden begint, doet u er goed aan de datasheet van de controller te raadplegen. Solomon Systech doet hier nogal geheimzinnig over — onder [2] vindt u een redelijk actuele versie (tenminste op met moment dat dit nummer ter perse gaat).



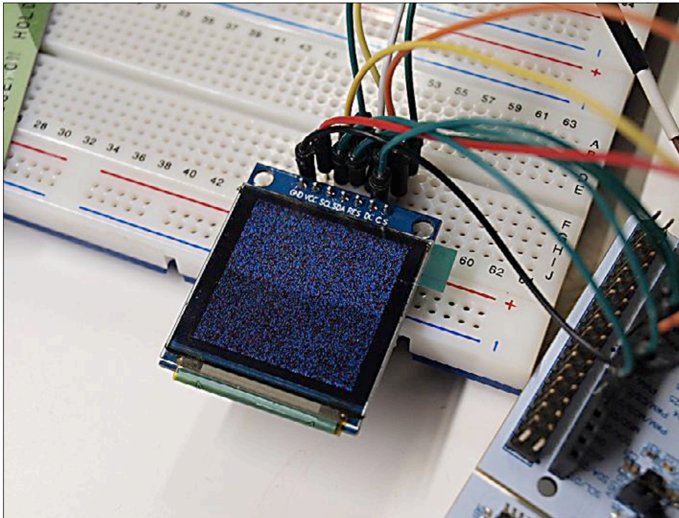
Figuur 2. De SSD1351 lijkt wat betreft de interne opbouw op zijn voorgangers.



Figuur 3. Deze instellingen zijn voor ons display snel genoeg.



Figuur 4. Het aansluiten van de kleuren-OLEDs is niet echt moeilijk. Het display kan worden geconfigureerd voor I²C- en SPI-aansturing.



Figuur 5. De informatie op het scherm is puur willekeurig omdat het werkgeheugen niet is geïnitialiseerd.

CUBE is — ondanks de zeer gecompliceerde bediening — een van de betere codegeneratoren in de embedded sector. Het “opnieuw overschrijven” van projecten na wijzigingen in het .ioc-bestand blijkt in de praktijk geen probleem te zijn — tenminste als u uw eigen code alleen in de sectoren van de gebruikerscode plaatst. In het geval van het toegangspunt zijn er twee beschikbaar, die er als volgt uitzien:

```
int main(void) {
. . .
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
   while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```

ST plaatst de `while`-lus — zowel het begin als het einde — onder de verantwoordelijkheid van de ontwikkelaar. In de volgende stap willen we beginnen met de “initialisatie”. Het is nuttig gebleken om een complete reset uit te voeren na de eerste start van het display.

Hiervoor maken we de reset-pin even laag, wachten we 500 ms en geven we het display nog eens 500 ms om zichzelf te initialiseren:

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */

HAL_GPIO_WritePin(DISPLAY_RESET_GPIO_Port, DISPLAY_
   RESET_Pin, GPIO_PIN_SET);
HAL_Delay(500);
HAL_GPIO_WritePin(DISPLAY_RESET_GPIO_Port, DISPLAY_
   RESET_Pin, GPIO_PIN_RESET);
HAL_Delay(500);
HAL_GPIO_WritePin(DISPLAY_RESET_GPIO_Port, DISPLAY_
   RESET_Pin, GPIO_PIN_SET);
HAL_Delay(500);
```

Vervolgens hebben we twee functies nodig, die elk één byte naar de display-controller schrijven. In de eerste stap gebruiken we de API `HAL_GPIO_WritePin`, die interactie met GPIO’s onder CUBE mogelijk maakt. Zowel `DISPLAY_DC_GPIO_Port` als `DISPLAY_DC_Pin` zijn constanten die het resultaat zijn van de ingevoerde ‘vriendelijke’ naam van de betreffende pin:

```
/* USER CODE BEGIN 0 */

void writeCommand(char _q) {
  HAL_GPIO_WritePin(DISPLAY_DC_GPIO_Port, DISPLAY_DC_
   Pin, GPIO_PIN_RESET);
  HAL_SPI_Transmit(&hspi1,&_q,1,100);
}

void writeData(char _q) {
  HAL_GPIO_WritePin(DISPLAY_DC_GPIO_Port, DISPLAY_DC_
   Pin, GPIO_PIN_SET);
  HAL_SPI_Transmit(&hspi1,&_q,1,100);
}
/* USER CODE END 0 */
```

De feitelijke overdracht van de gegevens vindt plaats via `HAL_SPI_Transmit`. De functie, zoals gebruikelijk in de ST-wereld, verwijst naar een poortobject dat wordt gegenereerd door een van de door CUBE gegenereerde methoden. Deze wordt gevolgd door een verwijzing naar een char-veld, de hoeveelheid gegevens die moet worden overgedragen en een time-out (die speelt hier geen grote rol). Door ontwikkelaars gecreëerde functies moeten zich in het bereik `USER CODE BEGIN 0` bevinden.

Stapsgewijze configuratie

Als u een complexe chip gebruikt, is het een goed idee om eerst te zoeken naar een referentie-implementatie. Een bijzonder dankbare bron hiervoor is de firma Adafruit, die onder [3] een kant-en-klare driver aanbiedt.

Daar vindt u onder andere een groep constanten, die de afzonderlijke registers van de chip definiëren — kopieer ze naar *main.h*. In de volgende stap beginnen we met de eigenlijke initialisering:

```
int main(void)
{
. . .
  HAL_Delay(500);

  writeCommand(SSD1351_CMD_COMMANDLOCK);   // set
   command lock
  writeData(0x12);
  writeCommand(SSD1351_CMD_COMMANDLOCK);   // set
   command lock
  writeData(0xB1);

  writeCommand(SSD1351_CMD_DISPLAYOFF);   // 0xAE
  . . .
  writeCommand(SSD1351_CMD_PRECHARGE2);
```

Write Data			Data bus																	
Bus width	Color Depth	Input order	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
8 bits/Serial	65k	1st	X	X	X	X	X	X	X	X	X	X	C_4	C_3	C_2	C_1	C_0	B_5	B_4	B_3
		2nd	X	X	X	X	X	X	X	X	X	X	B_2	B_1	B_0	A_4	A_3	A_2	A_1	A_0
8 bits/Serial	262k	1st	X	X	X	X	X	X	X	X	X	X	X	X	C_5	C_4	C_3	C_2	C_1	C_0
		2nd	X	X	X	X	X	X	X	X	X	X	X	X	B_5	B_4	B_3	B_2	B_1	B_0
		3rd	X	X	X	X	X	X	X	X	X	X	X	X	A_5	A_4	A_3	A_2	A_1	A_0
16 bits	65k		X	X	C_4	C_3	C_2	C_1	C_0	B_5	B_4	B_3	B_2	B_1	B_0	A_4	A_3	A_2	A_1	A_0
16 bits	262k format 1	1st	X	X	X	X	X	X	X	X	X	X	X	X	C_5	C_4	C_3	C_2	C_1	C_0
		2nd	X	X	X	X	B_5	B_4	B_3	B_2	B_1	B_0	X	X	A_5	A_4	A_3	A_2	A_1	A_0
16 bits	262k format 2	1st	X	X	X	X	$C1_5$	$C1_4$	$C1_3$	$C1_2$	$C1_1$	$C1_0$	X	X	$B1_5$	$B1_4$	$B1_3$	$B1_2$	$B1_1$	$B1_0$
		2nd	X	X	X	X	$A1_5$	$A1_4$	$A1_3$	$A1_2$	$A1_1$	$A1_0$	X	X	$C2_5$	$C2_4$	$C2_3$	$C2_2$	$C2_1$	$C2_0$
		3rd	X	X	X	X	$B2_5$	$B2_4$	$B2_3$	$B2_2$	$B2_1$	$B2_0$	X	X	$A2_5$	$A2_4$	$A2_3$	$A2_2$	$A2_1$	$A2_0$
18 bits	262k		C_5	C_4	C_3	C_2	C_1	C_0	B_5	B_4	B_3	B_2	B_1	B_0	A_5	A_4	A_3	A_2	A_1	A_0

Figuur 6. De SSD1351 is zeer flexibel

```
  writeData(0x01);
  writeCommand(SSD1351_CMD_DISPLAYON);
```

De ingekorte code hierboven stelt een reeks parameters in om de configuratie van de chip te beïnvloeden. Tenslotte verzendt het de opdracht `SSD1351_CMD_DISPLAYON`, waarmee de controller wordt aangespoord om het scherm in te schakelen. Dergelijke commando's zijn onder meer nuttig omdat ze het in- en uitschakelen van het display mogelijk maken zonder gebruik van transistoren en dergelijke.
Nu kunnen we het programma uitvoeren. Het produceert het scherm dat wordt weergegeven in **figuur 5**.

En dan nu in kleur

Het weergeven van 'willekeurige' patronen kan grappig zijn, maar heeft in de praktijk weinig zin. Het zou veel leuker zijn om betekenisvolle informatie op het scherm te zetten. De SSD1351 kent twee kleurenresolutiemodi volgens het gegevensblad: *262k color* geeft meer kleurdetails, maar heeft 18 bit nodig (6:6:6). *65k color* werkt met slechts 16 bit per pixel (5:6:5). Bovendien kent de controller een groot aantal verschillende formaten die via de tabs kunnen worden geselecteerd. Hieronder willen we met het type *16bits/65K* werken; de tabel in **figuur 6** toont de andere mogelijkheden.

Nu kunnen we beginnen met de uitvoer van een constante waarde:

```
while (1) {
    writeCommand(SSD1351_CMD_SETCOLUMN);
    writeData(0x00);
    writeData(0x7F);
    writeCommand(SSD1351_CMD_SETROW);
    writeData(0x00);
    writeData(0x7F);
    writeCommand(SSD1351_CMD_WRITERAM);

    for (uint16_t i=0; i < 128*128; i++) {
      writeData(0x00);
      writeData(0xFF);
    }
    HAL_Delay(10);
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
}
```

Deze code is zeker niet erg effectief — hij loopt het hele beeldgeheugen door en levert twee pixels voor elke waarde. In de praktijk hadden we dat gedoe met de CS-pin niet nodig, maar we hebben geen zin om het programma op dit punt te veranderen. Na het uitvoeren van het programma ziet u een constante kleur op het scherm.

Vervolgens moeten we het eigenlijke schermgeheugen aanmaken — de Engelse benaming hiervoor is *framebuffer*. Omdat de geheugenbronnen van onze controller niet onbeperkt zijn, definiëren we in de eerste stap een speciaal geheugenveld:

```
#pragma pack(2)
typedef struct {
  unsigned int rField : 5;
  unsigned int gField: 6;
  unsigned int bField: 5;
} colorquine;
```

De ervaring leert dat directe evaluatie van de inhoud van bitvelden soms veel werk voor de processor betekent. Een snelle manier — met niet te veel overhead als het scherm niet al te vaak vaak wordt ververst — om het probleem te omzeilen, is om gebruik te maken van bitmanipulatie:

```
char makeLower(colorquine _x){
    char work = _x.rField<<3;
    char temp = _x.gField>>3;
    return work+temp;
}
char makeHigher(colorquine _x){
    char work= (_x.gField & 0b111)<<5;
    return work + _x.bField;
}
```

Zowel `makeLower` als `makeHigher` hebben de taak om de drie

kleurwaarden naar een formaat te vertalen dat de controller begrijpt. Hier willen we makeHigher bespreken: in de eerste stap maskeren we de waarde van .gField om het relevante deel vast te leggen (lees: de laatste drie bits). Deze gaan dan naar links door middel van een verschuiving om de blauwe kleurwaarde door optelling toe te voegen.
Nu ontbreekt alleen nog de uitvoer van de geleverde informatie:

```
for (uint16_t i=0; i < 128*128; i++) {
    writeData(makeLower(myDisplayStorage[i]));
    writeData(makeHigher(myDisplayStorage[i]));
}
```

Nu kunnen we eigenlijk alles doen wat we willen: pak uw favoriete grafische algoritme en laat dat los op de framebuffer. Als u geheugen wilt besparen, kunt u als alternatief ook een kleurendisplay met vast kleurenpalet gebruiken.

En het stroomverbruik...

OLED werd — na introductie op de mobiele markt — beschouwd als een energiebesparend wonder. Dit vraagt erom het stroomverbruik te meten. We hebben hier twee opties voor.
Ten eerste de jumper zoals beschreven door professor Dogan Ibrahim in het STM32-boek (zie het kader “In de store”). En ten tweede het tussenschakelen van een multimeter. De auteur koos voor de tweede optie; gebruikt werd een Solartron 7150.

Dure adapterprint!

Wilt u een grotere serie plannen, doet u er goed aan de “dochterboards” zelf te ontwikkelen. De Chinese postorderbedrijven willen hier graag een graantje meepikken en verlangen hoge marges.

Het stroomverbruik van OLED’s is afhankelijk van de situatie, zoals weergegeven in **tabel 1**.

Samengevat

Wanneer het display met 3,3 V wordt gevoed, is de helderheid niet om over naar huis te schrijven — buiten was het uit China geïmporteerde display eigenlijk onbruikbaar. Bij toepassingen binnenshuis zijn de schermen echter goed leesbaar.
In de tijd van klassieke PDA’s hadden eindgebruikers al bemerkt dat apparaten met kleurenschermen als ‘duurder’ ervaren werden dan hun monochrome tegenhangers. Wie zich dus in het hoger geprijsde of status-segment van de markt wil begeven, doet er goed aan tenminste enioge aandacht aan het aanbod te schenken. ⏮

(180346-02)

Tabel 1. Stroomverbruik.

Weergegeven kleur resp. modus	Stroomverbruik

myDisplayStorage[i].rField=0;
myDisplayStorage[i].gField=0;
myDisplayStorage[i].bField=0;
	4,2 mA

myDisplayStorage[i].rField=0b11111;
myDisplayStorage[i].gField=0;
myDisplayStorage[i].bField=0;
	67 mA

myDisplayStorage[i].rField=0;
myDisplayStorage[i].gField=0b111111;
myDisplayStorage[i].bField=0;
	57 mA

myDisplayStorage[i].rField=0;
myDisplayStorage[i].gField=0;
myDisplayStorage[i].bField=0b11111;
	102 mA

myDisplayStorage[i].rField=0b11111;
myDisplayStorage[i].gField=0b111111;
myDisplayStorage[i].bField=0b11111;
	112 mA

DISPLAYOFF	1,3 mA



IN DE STORE

→ STM32 Nucleo L476RG Board
www.elektor.nl/stm32-nucleo-l476rg-board

→ Boek „Programming with STM32 Nucleo Boards“
+ gratis STM32 Nucleo L476RG Board
www.elektor.nl/programming-with-stm32-nucleo-boards



Weblinks

[1] Software-download: www.elektormagazine.nl/180346-02

[2] Datasheet van de OLED-controller SSD1351: www.newhavendisplay.com/app_notes/SSD1351.pdf

[3] OLED-configuratie: https://github.com/adafruit/Adafruit-SSD1351-library/blob/master/Adafruit_SSD1351.h

HOMELAB PROJECT

LED-dimmers (2)

het hoeft niet altijd digitaal te zijn

Roel Arits (Nederland)

Van de vele analoge LED-dimmers die Roel Arits naar aanleiding van onze kleine Elektor Labs-uitdaging [1] heeft ingezonden, hebben we er in het vorige nummer [2] al twee in de schijnwerpers gezet. In deze tweede en laatste aflevering laten we er nog twee de revue passeren.

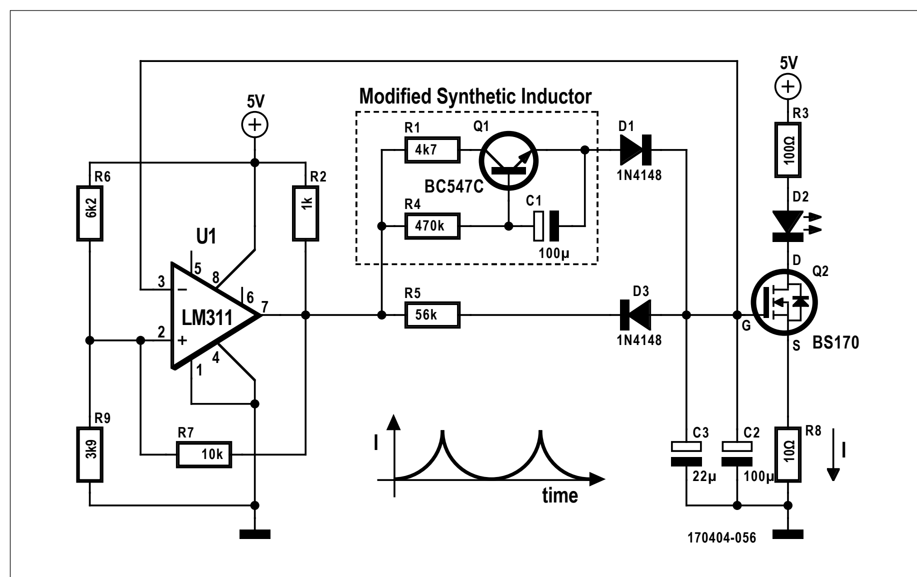
Eigenschappen

- Verschillende (analoge) manieren om een LED te dimmen
- Alleen standaardcomponenten
- Voer voor experimentatoren

In de vorige aflevering hebben we al opgemerkt dat we, voor een waargenomen lineaire toe- en afname van de helderheid van een LED (of gloeilamp, maar we beperken ons hier tot LED's) de stroom door die LED niet lineair moeten laten toe- en afnemen, maar exponentieel. Dat heeft alles te maken met de min of meer logaritmische gevoeligheidskarakteristiek van onze ogen, zoals al in de 19de eeuw in de wet van Weber-Fechner [3] is beschreven. Daarover hebben we het in de vorige aflevering al uitgebreid gehad, zodat we hier volstaan met de opmerking dat we een exponentieel toe- of afnemende stroom door een LED nodig hebben om deze logaritmische karakteristiek te compenseren zodat we de helderheid lineair zien veranderen.

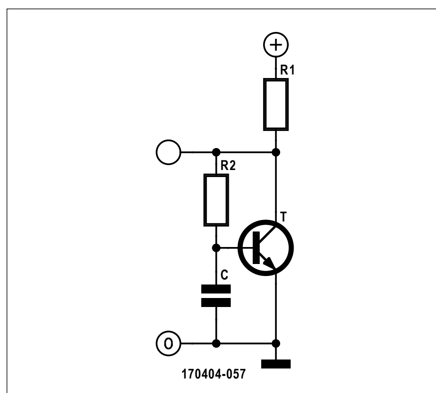
Gyrator

Er wordt wel beweerd dat alle wegen naar Rome leiden — en minstens even zoveel leiden naar een analoge LED-fader. Bij de hier beschreven variant wordt gebruik gemaakt van een relaxatie-oscillator, opgebouwd rond een als Schmitt-trigger geschakelde opamp (zie het schema van **figuur 1**).
Comparator IC1 vormt het hart van de schakeling. De uitgangsspanning ervan wordt via weerstand R7 teruggekoppeld naar de niet-inverterende ingang; hierdoor wordt een zekere hysterese gecreëerd rond de drempelspanning (die met spanningsdeler R6/R9 wordt ingesteld). Op deze manier verandert de comparator in een Schmitt-trigger; de waarde van R7 is bepalend voor de amplitude van het uitgangssignaal.
De timing van de oscillator wordt bepaald door het op- en ontladen van de parallelgeschakelde condensatoren C2 en C3. Door de beide diodes D1 en D3 worden de op- en ontlaadpaden strikt van elkaar gescheiden.
Het ontladen is geen probleem: dit gaat via R5 en op die manier krijgen we een ruwweg exponentieel afnemende spanning over de condensatoren — precies wat we nodig hebben.
Maar nu het opladen van de condensatoren — we willen de spanning over de elco's ook graag exponentieel laten toenemen. Daarvoor hebben we een schakeling nodig die bij een constante spanning de stroom exponentieel laat toenemen. Wanneer we een gewone spoel nemen, zal de stroom door die spoel bij het aanleggen van een vaste spanning logaritmisch toenemen. Hetzelfde gebeurt wanneer we in plaats van een echte spoel een gyrator gebruiken — dus een halfgeleiderschakeling die het gedrag van een spoel nabootst. Maar een gyrator kan gemakkelijker dan een echte spoel zodanig worden gemodificeerd dat de spanning een exponentieel verloop krijgt. En dat is precies wat de auteur hier heeft gedaan.
Transistor T1 met omringende onderdelen (R1, R2 en C1) vormt een gesimuleerde spoel (zelfinductie); de modificatie bestaat uit weerstand R1 (daar komen

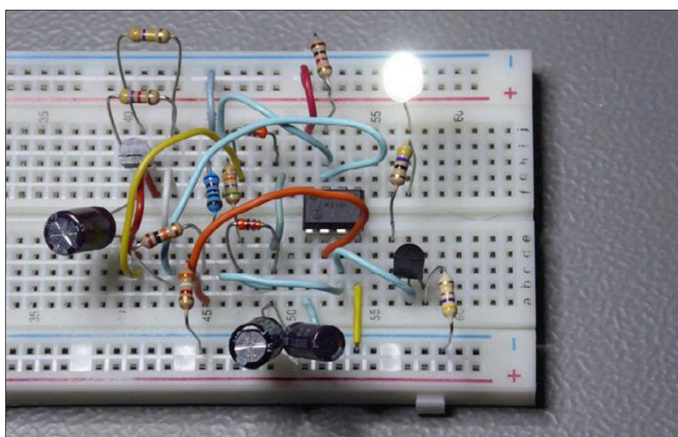


Figuur 1. Het schema van de analoge LED-dimmer met gemodificeerde gyrator.

we zometeen op terug).
In **figuur 2** hebben we de hoofdrolspelers van de gyrator separaat in het zonnetje gezet. Het voert te ver om hier heel diep op de theorie achter de gyrator in te gaan (daarvoor verwijzen we naar de literatuur, bijvoorbeeld [4]). Het gedrag kan echter kwalitatief worden begrepen wanneer we de gyrator in het frequentiedomein bekijken, en wel bij lage en hoge frequenties.
In het eerste geval (zeer lage frequentie of gelijkspanning op de ingang) zal condensator C een zeer hoge impedantie vormen. De transistor krijgt dan via R2 basisspanning en is in geleiding; gezien 'in' de ingang vormt de schakeling nagenoeg een kortsluiting — precies zoals we bij een echte spoel zouden verwachten.
In het andere geval (hoge frequenties) vormt de condensator een kleine impedantie en 'steelt' als het ware basisspanning van de transistor. Die geleidt dan minder en in de ingang gezien zien we een hogere impedantie — wederom: net als bij een echte spoel.
De modificatie met R1 (in het schema van figuur 1) zorgt ervoor dat de spanning op de collector van T1 inzakt naarmate de collectorstroom (en dus ook de emitterstroom) toeneemt. Op die manier worden C2/C3 met een ongeveer exponentieel toenemende stroom geladen.
De rest van de schakeling is standaard: de spanning over de condensatoren wordt gebufferd door MOSFET T2 en door weerstand R8 omgezet in een stroom door de LED D2 (maximaal ongeveer 5 mA). R3 vormt een extra stroombegrenzing.
Met de gegeven dimensionering bedragen de fade-in- en fade-out-tijden ongeveer 1,2 s (toe- en afname van de helderheid van de LED verlopen ongeveer symmetrisch).
Wanneer u de minimale helderheid van de LED instelbaar wilt maken, kunt u R6 en R9 vervangen door een (instel)potmeter van 10 kΩ, waarbij de loper van die potmeter naar R7 en de niet-inverterende ingang van de comparator gaat.
Het verdient aanbeveling voor de LED een high-efficiency-type te nemen, omdat die bij kleine stromen een hogere lichtopbrengst heeft dan een gewone LED.



Figuur 2. Wanneer we de gyrator uit de schakeling lichten, is de werking beter te doorgronden.



Figuur 3. Het prototype van de auteur op breadboard.

Deze schakeling heeft een zwaarwegend nadeel (u voelde 'm waarschijnlijk al aankomen): de dimfrequentie (fade-in en fade-out) kan niet eenvoudig met een enkele potmeter worden gewijzigd. Daarvoor moeten verschillende componenten tegelijk van waarde veranderen: samen met C2/C3 moet ook C1 worden gevarieerd (en eventueel ook R5).
De foto van **figuur 3** toont de opstelling van de auteur op breadboard.



IN DE STORE

→ Boek: Elektronica Kunst en Kunde 1
www.elektor.nl/elektronica-kunst-en-kunde-1

→ Boek: Basiscursus Elektronica
www.elektor.nl/basiscursus-elektronica

En als klap op de vuurpijl...

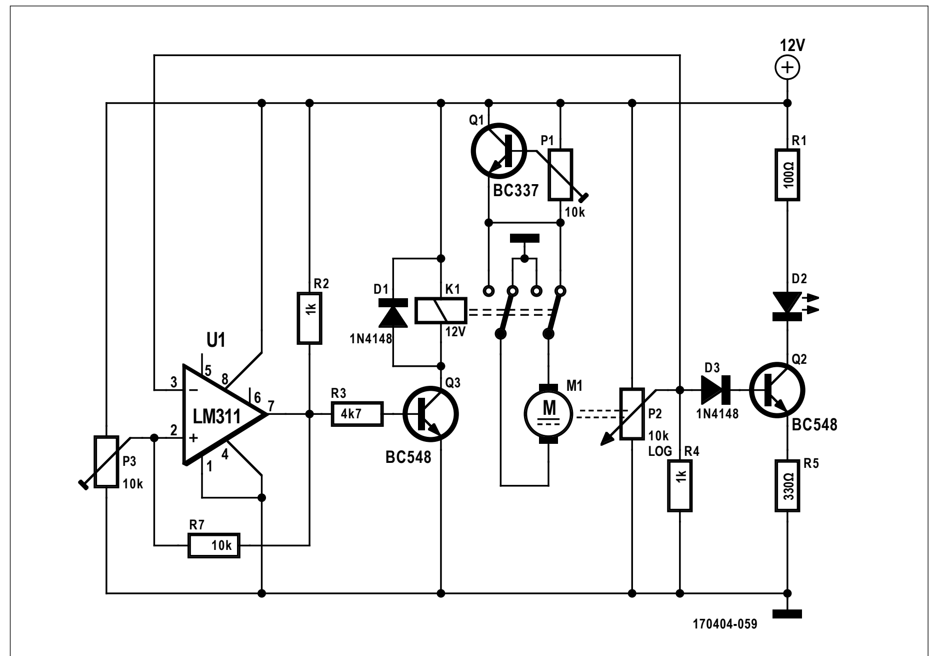
...een mechanische dimmer. Hierbij wordt gebruik gemaakt van een gemotoriseerde potmeter in combinatie met een elektromechanisch relais — en het uiteindelijke effect is zo retro dat het bijna steampunk genoemd kan worden...
In moderne 'low cost' audioversterkers zien we doorgaans geen 'echte' volumeregelaar meer — het volume wordt geheel elektronisch geregeld. Dat maakt het eenvoudiger om met weinig meerkosten een afstandbediening te realiseren. In versterkers uit het hogergeprijsde segment, en zeker in high-end apparatuur, treffen we echter hoogwaardige potentiometers aan (die minder vervorming en ruis introduceren), en om afstandsbediening mogelijk te maken zijn deze potmeters gemotoriseerd. De bekendste voorbeelden zijn waarschijnlijk de motorpotmeters van Alps, die een uitstekende kwaliteit voor relatief weinig geld bieden.
De auteur heeft voor zijn mechanische LED-dimmer natuurlijk niet zijn nieuwe stereoversterker geplunderd, maar een motorpotmeter uit een oudere versterker gesloopt — geen modern stofdicht exemplaar maar een potmeter met aangebouwde vertraging en elektromotor. Een slipkoppeling tussen vertraging en potmeteras maakt het mogelijk het volume ook met de hand te regelen.
In audioversterkers worden bijna altijd logaritmische potmeters gebruikt, dit vanwege de logaritmische gevoeligheidskarakteristiek van ons gehoor (niet alleen onze ogen maar ook onze oren — en eigenlijk al onze zintuigen — voldoen aan de wet van Weber-Fechner). Wanneer we een logaritmische motorpotmeter gebruiken om de intensiteit van een LED te regelen en de motor met een constante snelheid laten draaien, dan krijgen we haast vanzelf de gewenste exponentiële helderheidsvariatie die ons oog als lineair ervaart.
Terzijde: wanneer u met deze mechanische dimmer wilt experimenteren maar toevallig alleen een lineaire motorpotmeter hebt liggen, dan kunt u die eenvoudig een min of meer logaritmische karakteristiek geven door een vaste weerstand parallel aan een van de potmeter-

helften te zetten. Op die manier kan ook de karakteristiek van een logaritmische potmeter eventueel worden aangepast (zie weerstand R4 in het schema van **figuur 4**).

Aan elektronica hebben we dus een schakeling nodig die de motor van de potmeter met een constante snelheid linksom en rechtsom laat draaien — en dat is niet zo bijster ingewikkeld. Het hart van de schakeling wordt (weer) gevormd door een als Schmitt-trigger geschakelde comparator (IC1 in figuur 4). De spanning op de loper van de potmeter wordt vergeleken met een met instelpotmeter P3 instelbare referentiespanning; door de terugkoppeling van de uitgang naar de niet-inverterende ingang (via weerstand R6) wordt een hysterese rond die referentiespanning verkregen waardoor twee schakeldrempels ontstaan. De uitgangsspanning van de comparator bepaalt (via het elektromechanische relais) de draairichting van de motor: de motor draait in de ene richting tot de betreffende drempelspanning is bereikt; dan keert de draairichting om en draait de motor in de andere richting tot de tweede drempelspanning is bereikt. Dit is dus een mechanische oscillator; de spanning op de loper van de potmeter bepaalt (via transistor T2) de helderheid van de LED.

Figuur 5 geeft een impressie van de schakeling op breadboard; de ouderwetse motorpotmeter is hier voorzien van een bepaald niet stofdichte vertraging! In het oscillogram van **figuur 6** kunt u zien dat de schakeling van de auteur een fraai exponentieel verlopende spanning levert. ◀

(170404-B-01)



Figuur 4. De mechanische LED-dimmer is opgebouwd rond een motorpotmeter.

Waarschuwing

De in dit artikel (en in de voorgaande aflevering) beschreven schakelingen zijn *proof of concept*-ontwerpen, bedoeld voor laagspanningsvoeding (5...12 V) en voor de aansturing van een enkele LED. De schakelingen zijn **absoluut niet** geschikt voor het regelen van LED-lampen en al **helemaal niet** voor aansluiting op het lichtnet!

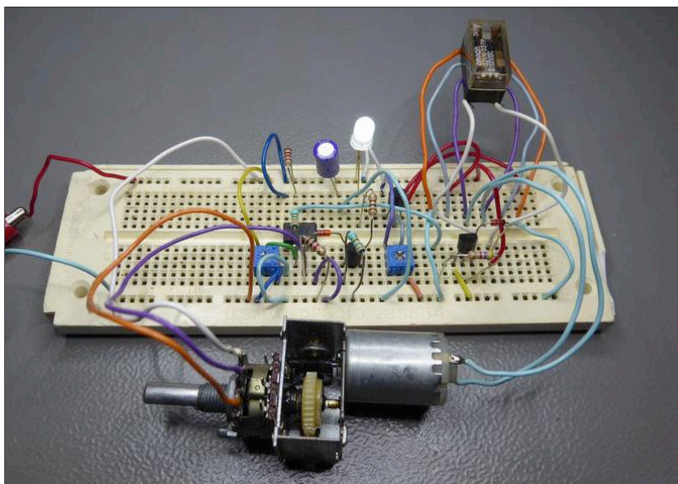
Weblinks

[1] De winnaars van de Labs-wedstrijd: https://goo.gl/JWo8oC

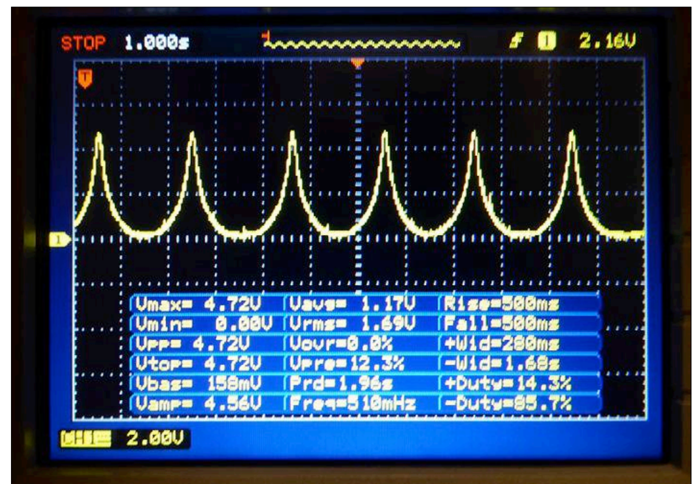
[2] Het eerste deel van deze tweedelige serie: www.elektormagazine.nl/170404-01

[3] Wet van Weber-Fechner: https://nl.wikipedia.org/wiki/Wet_van_Weber

[4] Elektronica Kunst en Kunde deel 1 (analoge technieken): www.elektor.nl/elektronica-kunst-en-kunde-1



Figuur 5. Dankzij de 'antieke' motorpotmeter maakt de mechanische LED-dimmer een fraaie retro-indruk.



Figuur 6. Oscillogram van de spanning op de loper van de motorpotmeter.

Kortsluiting-zoeker

met stroombron en Hall-sensor



Josef Tausch (Duitsland)

Dit testapparaat lokaliseert een kortsluiting door een (gepulste) teststroom door de verdachte leiding te sturen en vervolgens het hierdoor veroorzaakte magnetische veld te detecteren.

Als in een schakeling of kabel een kortsluiting optreedt, is het niet zo eenvoudig om de plaats ervan te lokaliseren. Het lijkt een goed idee om het magnetisch veld van de kortsluitstroom te volgen, maar dat zal niet lukken als door de kortsluiting het onderdeel dat de stroom levert wordt overbelast. Het zal dan tijdelijk (of permanent) niet meer functioneren waardoor er helemaal geen stroom loopt. Het testapparaat moet dus niet alleen in staat zijn om een magnetisch veld te detecteren, maar ook om zelf een teststroom te genereren die bij een kortsluiting stabiel door de leiding loopt. De kortsluitingzoeker kan hierbij op twee standen worden ingesteld: gelijkstroom of stroompulsen. Bij gelijkstroom kan het magnetisch veld al met een zeer kleine beweging van de Hall-sensor worden gedetecteerd. In de puls-modus kan bovendien het door de stroom opgewekte magnetisch veld niet worden verwisseld met andere (storende) magnetische velden zoals het aardmagnetisch veld.
Het testapparaat bestaat uit twee onafhankelijke delen die in aparte behuizingen zijn ondergebracht: een stroombron die de teststroom door de kortgesloten leiding laat lopen en een sensoreenheid die het magnetisch veld detecteert en de aanwezigheid ervan optisch en akoestisch weergeeft.

Beide delen beschikken over hun eigen onafhankelijke voeding in de vorm van een lithium-ionaccu en een laad-IC. Het testapparaat maakt het lokaliseren van ‘echte’ laagohmige kortsluitingen mogelijk zonder dat er onderdelen hoeven te worden verwijderd. Met de sensoreenheid kunnen niet alleen kortsluitingen worden gelokaliseerd, maar kunnen ook magnetische velden van andere oorsprong (bijvoorbeeld afkomstig van relais of elektromagneten) worden aangetoond.

Orde van grootte

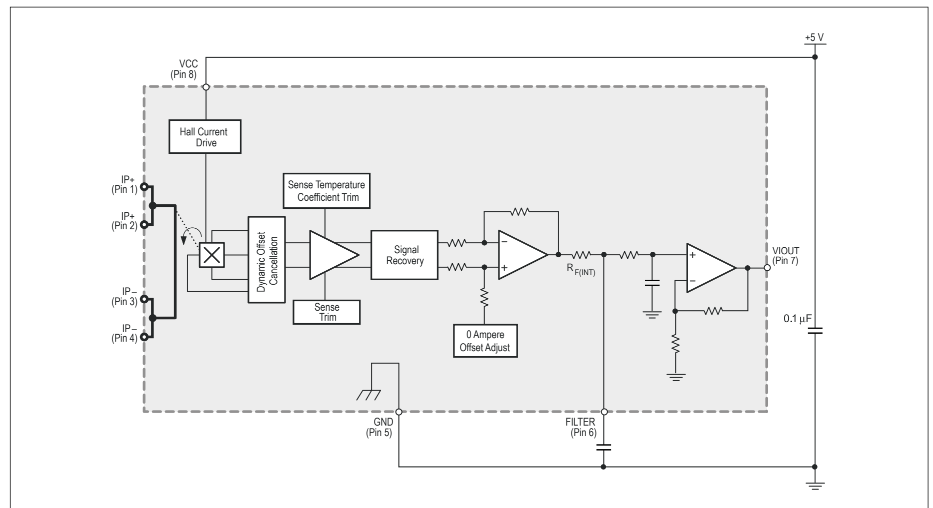
Voordat er met het ontwerpen van de elektronica kon worden begonnen, moest eerst duidelijk zijn met welke magnetische veldsterktes we hier te maken hebben. De zeer geringe gevoeligheid van Hall-sensoren, die in de orde van grootte van 10...100 mV/mT ligt, kan namelijk een probleem opleveren.

De sterkte van een magnetisch veld H kan via de magnetische fluxdichtheid B in de eenheid T (tesla) worden gemeten. Om enig gevoel te krijgen voor de orde van grootte volgen hier een paar voorbeelden: het aardmagnetisch veld heeft op onze breedtegraad (nog) een fluxdichtheid van ongeveer 50 µT; de grenswaarde voor door het 50-Hz-stroomnet veroorzaakte elektromagnetische stoorvelden ligt bij 100 µT.
Deze waarden kunnen we vergelijken met de magnetische flux B die een stroom I in een rechte geleider veroorzaakt:

$$B = \mu_0 * H = \mu_0 * I/2\pi r$$

Hierin is μ_0 de magnetische permeabiliteit van het vacuüm en r de afstand tot de geleider.

We kunnen nu uitrekenen dat een stroom van 250 mA een veld veroorzaakt dat op een afstand van 3 mm van de geleider slechts 16,7 µT bedraagt. Het aardmagnetisch veld is drie keer zo sterk, om nog maar te zwijgen over de 50-Hz-stoorvelden! Het komt er dus op aan om een zo gevoelig mogelijke sensor te vinden en deze zo dicht mogelijk bij de geleider te plaatsen. Tegelijkertijd moet de teststroom zo groot mogelijk worden gekozen, maar nog wel acceptabel voor een batterijgevoed apparaat. Alleen op deze manier kunnen we voorkomen dat het nuttige signaal volledig ‘in de ruis’ verdwijnt!



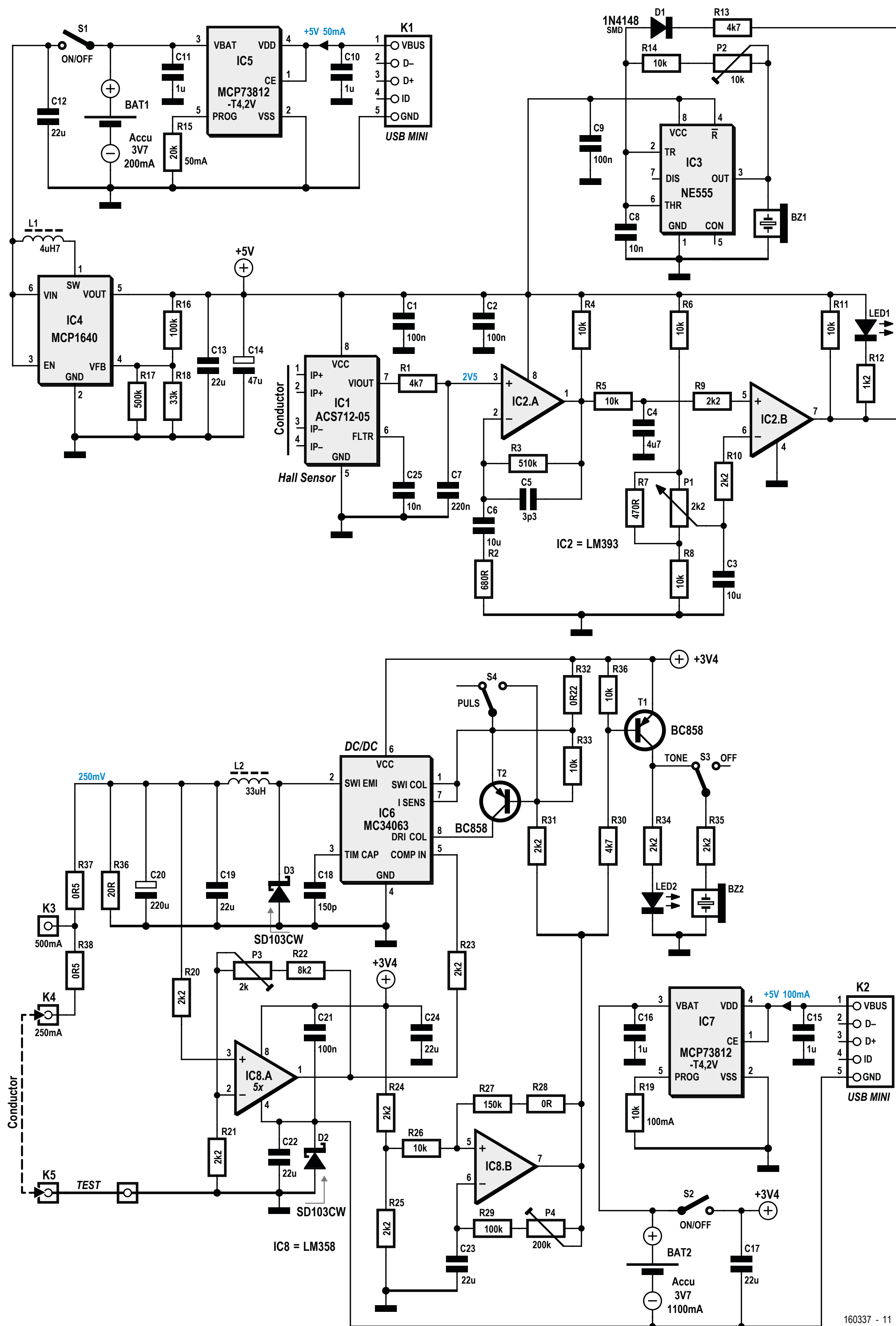
Figuur 1. Het ‘geamputeerde’ Hall-sensor-IC (afbeelding: Allegro MicroSystems).

De sensoreenheid

De eerste opgave bij het ontwikkelen van de schakeling was dus de keuze van een geschikte Hall-sensor. De auteur heeft verschillende Hall-sensoren uitgeprobeerd en is uiteindelijk bij de ACS712-05 van Allegro Microsystems [1] uitgekomen. Zoals in **figuur 1** is te zien, heeft dit SOIC acht aansluitingen. Hiervan zijn 1-2 en 3-4 verbonden met een interne stroomvoerende geleider waarvan het magnetisch veld direct in het IC wordt opgebouwd. Deze methode is voor onze kortsluitingzoeker ongeschikt omdat we de kortgesloten leiding niet willen onderbreken. Om ervoor te zorgen dat het testapparaat, of beter gezegd zijn sensorelement, zo dicht mogelijk bij de leiding kan worden gehouden, heeft de auteur zelfs de niet gebruikte pennen afgevijld!
Bij dezelfde fabrikant zijn ook Hall-sensoren in SIP-3-behuizing verkrijgbaar, zoals de A1389LUA-9T die een zodanig hoge gevoeligheid van 90 mV/mT heeft dat het IC niet met een vijl hoeft te worden bewerkt.
In **figuur 2** ziet u het volledige tweedelige schema van het testapparaat, met de sensoreenheid bovenin en de stroombron in het onderste gedeelte.
De uitgangsspanning van de sensor ACS712-05 (IC1) op pen 7 bedraagt zonder magnetisch veld +2,5 V en wordt afhankelijk van de stroomrichting hoger of lager. Omdat de spanningsverandering, die door R1 en C7 wordt afgevlakt, in ons voorbeeld (250 mA op 3 mm afstand) slechts enkele millivolt bedraagt, moet deze aanzienlijk worden versterkt en vervolgens met een met P1 instelbare drempelspanning (gevoeligheid) worden vergeleken. Voor deze taak is de klassieke dubbele comparator LM393 (IC2) bij uitstek geschikt. De uitgang van IC2.B levert een negatieve puls die niet alleen een LED aanstuurt maar ook een akoestische indicator triggert. U hoeft dan bij het testen niet voortdurend naar de LED te kijken. De akoestische indicator bestaat uit een eenvoudige 555-timer (IC3) en een passieve piëzo-buzzer. De frequentie kan met P2 worden ingesteld.
Ook de stroombron is voorzien van akoestische en optische signalering. Dit is erg belangrijk want in de pulsmodus zorgt deze ervoor dat u een concreet magnetisch veld volgt en niet door een stoorveld op een dwaalspoor wordt gebracht. De LED’s en de zoemers moeten namelijk tegelijkertijd of afwisselend oplichten respectievelijk geluid produceren. Als de LED’s en de piëzobuzzers afwisselend actief zijn, geeft dit aan dat het hier de tegengestelde stroomrichting in de retourleiding betreft.
De sensoreenheid wordt gevoed met 5 V. Omdat de lithium-ionaccu maar 3,7 V levert, zorgt een DC/DC-converter MCP1640 van Microchip [2] voor het opvoeren van de spanning. De uitgangsspanning wordt met een spanningsdeler (R16...R18) ingesteld volgens:

$$V_{OUT} = [(R16/R17||R18)+1]*1{,}21\ V$$

Er zijn verschillende varianten van deze converter, het maakt niet uit welke u gebruikt.
De sensoreenheid is net als een meetprobe in een eigen behuizing ondergebracht, die uit kunststof L-profielen (25



Figuur 2. Het schema van de kortsluitingzoeker met sensoreenheid (boven) en stroombron (onder).

mm x 25 mm) uit de bouwmarkt is samengesteld. De Hall-sensor bevindt zich in de tip die langs de stroomgeleider kan worden bewogen. De sensoreenheid wordt gevoed uit een accu die via een op het apparaat gemonteerde mini-USB-connector kan worden geladen.

De stroombron

Voor de stroom door de kortgesloten leiding is in het voorbeeld niet zonder reden een waarde van ongeveer 250 mA gekozen. Deze waarde lijkt een goed compromis te zijn tussen de sterkte van het magnetisch veld bij het sensorelement en de stroom waarmee je een batterijgevoede stroombron (en ook de leiding zelf) zonder problemen kunt belasten. Als door het sensorgedeelte geen eenduidig signaal wordt geleverd zijn er twee mogelijkheden, namelijk de stroomsterkte verhogen naar 500 mA en/of van gelijkstroom overschakelen naar stroompulsen. Met name die stroompulsen moeten ervoor zorgen dat het gegenereerde magnetisch veld niet meer met stoorvelden kan worden verwisseld.

Laten we eens kijken hoe dit allemaal elektronisch is gerealiseerd. De energie wordt geleverd door een LiIon-accu met een spanning van 3,7 V die aan een DC/DC-converter (IC6) van het type MC34063 [3] wordt toegevoerd. Het IC is hier als buck-regelaar geschakeld waarbij de uitgangsspanning op 250 mV is ingesteld. Meer is niet nodig omdat de spanning niet maatgevend is voor de sterkte van het magnetisch veld. Meestal wordt de uitgangsspanning ingesteld met de verhouding van de weerstandswaarden in een weerstandsdeler die de interne comparator met de uitgang en met massa verbindt. Omdat we in ons geval extra functies willen toevoegen gebeurt het hier op een wat ongebruikelijke manier.

Opamp IC8.A, een LM358, versterkt de uitgangsspanning van de converter van 250 mV tot 1,25 V en voert deze spanning via R23 toe aan de interne comparator van de converter. Deze spanning komt overeen met de interne referentie van 1,25 V zodat het IC deze spanning zal stabiliseren. De tweede helft van IC8 is als multivibrator geschakeld. De frequentie moet ongeveer 1 Hz bedragen en wordt met P4 ingesteld. De uitgang van de opamp is verbonden met twee schakeltransistoren. T1 stuurt in het 1-Hz-ritme een LED aan en ook een actieve piëzo-buzzer (die gelukkig met S3 het zwijgen kan worden opgelegd). Van groter belang is dat via R31 de andere schakeltransistor (T2) wordt geactiveerd. Hiermee wordt de collector van de interne stuurtransistor van IC6 alleen tijdens een puls met de ingangsspanning verbonden. Tijdens de pauze tussen twee pulsen blijft de stuurtransistor ‘in de lucht hangen’ zodat de converter geen uitgangsstroom meer kan leveren. Met S4 wordt tussen continu- en pulsbedrijf omgeschakeld.

Om ervoor te zorgen dat de LM358 als versterker kan werken wordt zijn accuvoeding iets ten opzichte van massa verhoogd. Hiervoor wordt de minpool van de accu niet rechtstreeks met massa verbonden, maar via schottkydiode D2. Over de schottkydiode valt een kleine maar toereikende spanning van 0,3 V. Als alternatief kan hier een dubbele rail-to-rail opamp worden gebruikt waardoor de schottkydiode niet nodig is. De minpool van de accu kan dan direct aan massa worden gelegd en de schottkydiode kan worden weggelaten.

Aan de uitgang van DC/DC-converter IC6 vinden we een vaste

gelijkspanning die via twee weerstanden met massa is verbonden. We hebben hier dus te maken met een echte stroombron. Volgens de wet van Ohm bedraagt de stroom 250 mA als de 'te testen leiding' aan het open einde van R38 (K4) wordt aangesloten, en 500 mA als de aansluiting tussen de twee weerstanden (K3) wordt gemaakt. Het andere einde van de leiding wordt via K5 met massa verbonden, en wel via een draadje met een lengte van van ongeveer 10 mm waarvan het magnetisch veld kan worden gebruikt om de sensoreenheid te testen.

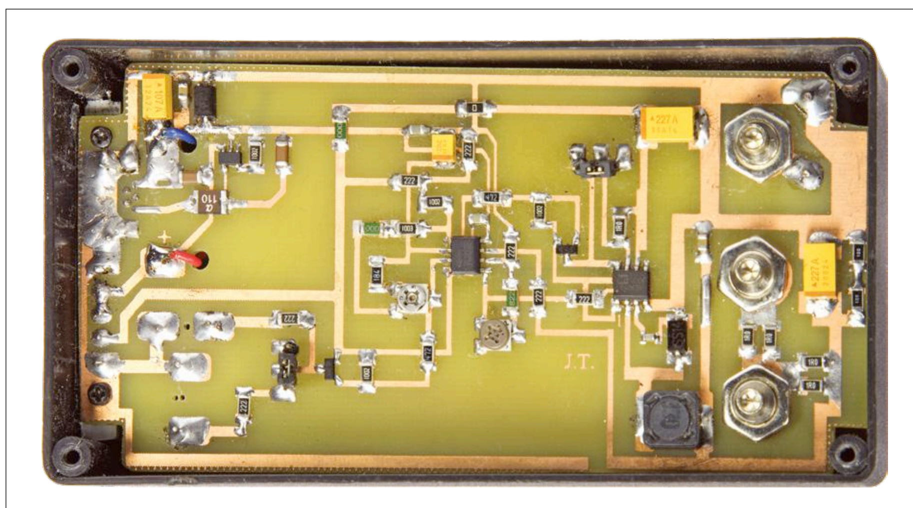
Accu's laden

Beide delen van het testapparaat worden door LiIon-accu's van energie voorzien. De grote accu (3,7 V/4,1 Wh) voor de stroombron is afkomstig uit een afgedankte Galaxy-3-smartphone van Samsung. Deze accu's, met typenummer EB504465VUCSTD, zijn voor een paar euro op het internet te vinden. De kleine accu (3,7 V/200 mAh) voor het sensorgedeelte (met de aanduiding D1A083382009) zat oorspronkelijk in een mini-DV-camcorder PMDV85. De auteur vond deze energiebron als goedkope restpost bij Pollin-Elektronik. Het exacte type is echter niet zo belangrijk, het gaat om de juiste spanning en afmetingen die passen bij de gekozen behuizing.

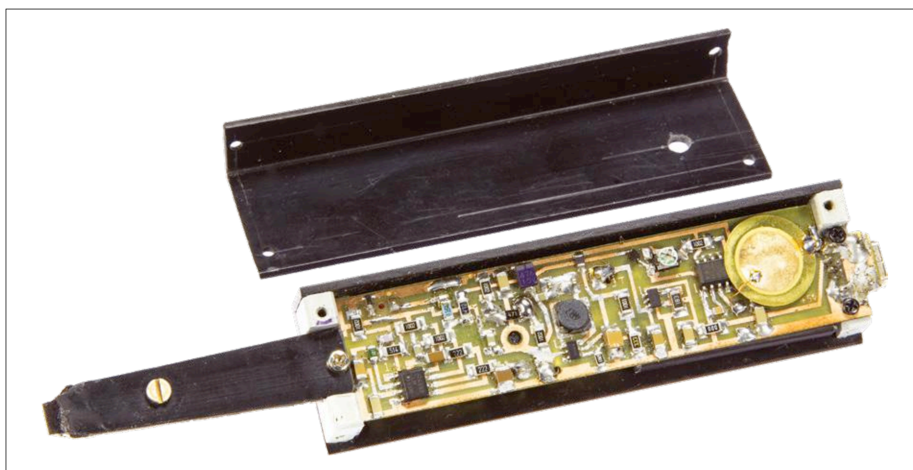
De accu's worden geladen met laadregelaars (IC5 en IC7) van het type MCP73812 [4] van Microchip dat speciaal is ontworpen voor het laden van dit soort accu's. De laadcurve is vast ingeprogrammeerd zodat u zich nergens zorgen over hoeft te maken. Het laadproces stopt niet automatisch als de accu volledig is geladen, maar pas als de voedingsspanning voor het laad-IC wordt uitgeschakeld. De laadstroom wordt ingesteld met een weerstand aan pen 5 van de regelaar.

De energiebehoefte van de stroombron is uiteraard afhankelijk van de belasting. Bij een continue teststroom van 500 mA (bij 250 mV) levert de 3,7-V-accu 148 mA. Bij de stroombron is de laadstroom met R19 = 10 kΩ op 100 mA (bij 5 V) ingesteld. Deze stroom kan zonder problemen door een USB-aansluiting worden geleverd.

Bij de sensoreenheid, die met ongeveer 25 mA beduidend minder energie nodig heeft, kan de laadstroom met R15 = 20 kΩ tot 50 mA worden beperkt. Ook hier wordt de energie geleverd door een USB-aansluiting..



Figuur 3. De stroombron past in een standaard kunststofbehuizing, waarin ook plaats is voor de platte accu.



Figuur 4. De behuizing van de sensoreenheid is opgebouwd uit kunststofprofielen van de bouwmarkt.

Opbouw en gebruik

De auteur heeft voor beide delen van het apparaat printlayouts ontworpen. U vindt deze op de projectpagina [5]. Voor de stroombron werd een passende standaard-kunststofbehuizing (**figuur 3**) gebruikt, voor de sensoreenheid moet u zelf een passende meetkopbehuizing bouwen. Zoals eerder vermeld gebruikte de auteur hiervoor simpele kunststofprofielen van de bouwmarkt (**figuur 4**).

Eventueel kan ook een kant-en-klaar product (zoals de LP1-behuizing van Teko) zodanig worden gemodificeerd dat de print er in past.

(160337)

Weblinks

[1] http://www.allegromicro.com/en/Sample-And-Buy/Contact-Sales.aspx#netherlands

[2] www.microchip.com/wwwproducts/en/en547080

[3] www.onsemi.com/pub/Collateral/MC34063A-D.PDF

[4] www.microchip.com/wwwproducts/en/MCP73812

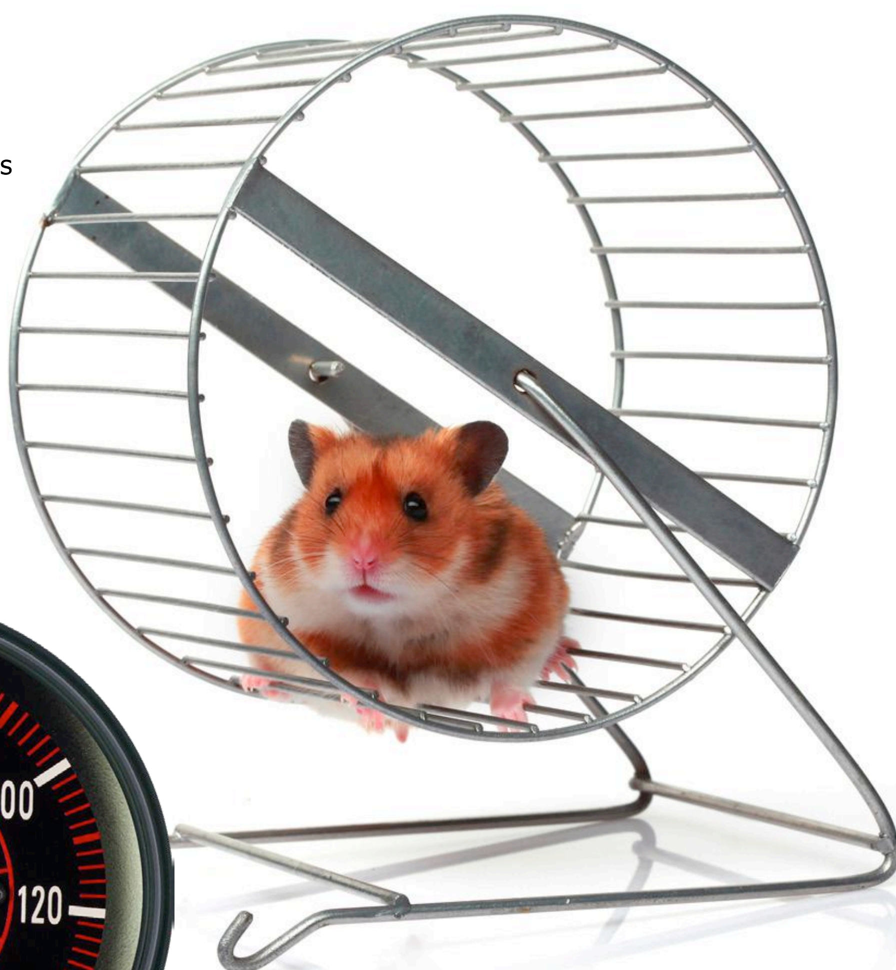
[5] www.elektormagazine.nl/160337

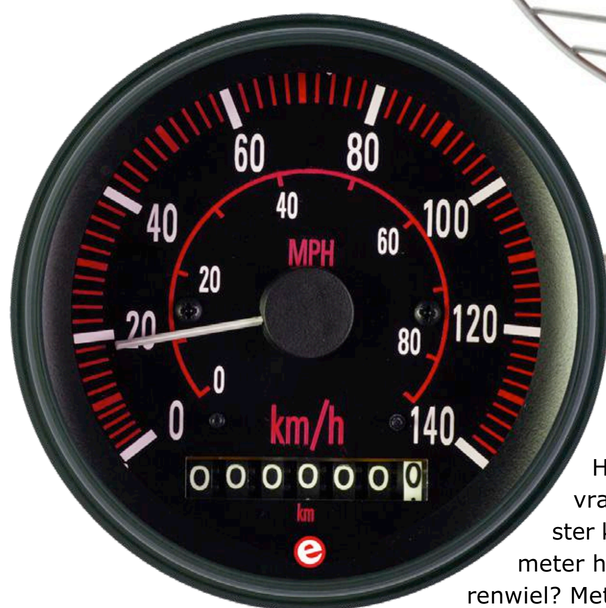
Hamster Run-O-Meter

km/h en afstand meten

Willem Tak

Bij menig knaagdier hangt er wel zoiets in de kooi: een wiel waarin ze naar hartenlust (of tot nachtelijke ergernis van het baasje) in kunnen rennen. Als elektronicus vráágt dat natuurlijk om experimenteren. Niet met het diertje (dat is de biotechnologie), maar met het bewegende wiel. In dit eenvoudige project loggen we de topsnelheid en de afgelegde weg.





Hebt u zich ooit afgevraagd hoe snel uw hamster kan rennen? Of hoeveel meter hij wel niet aflegt in dat renwiel? Met deze schakeling kunnen we niet alleen de topsnelheid meten, maar ook de totale afstand die de hamster aflegt tussen het begin en het einde van een meting. Natuurlijk wordt ook de duur van de meting geregistreerd en kunt u aan de hand van deze gegevens de gemiddelde snelheid berekenen.

Eigenschappen

- Logt afgelegde weg en maximum snelheid
- Achtergrondverlichting instelbaar
- Voeding 8-15 VDC
- 15 metingen worden opgeslagen in het EEPROM
- Eenvoudige bediening via draai-encoder

Opzet

Het idee was om niet (te veel) te rommelen met het kooi- of hamsterwiel. Om die reden plakken we twee IR-reflectiesensoren en een strookje reflecterend materiaal (tinfolie) op het wiel om omwentelingen te registreren. Twee IR-LED's dienen als de benodigde lichtbronnen. In de aanvankelijke opstelling gebruikten we slechts één sensor, maar al snel hebben we een tweede toegevoegd om meetfouten te voorkomen als de hamster het wiel alleen maar laat schommelen. Met twee sensoren registreren we volledige omwentelingen van de molen.

Hardware

De benodigde hardware staat afgebeeld in het schema van **figuur 1**. IC2 bestuurt de Hamster Run-O-Meter. Dit is een PIC18F26K22-microcontroller. In

PROJECT-INFO

Meten Loggen
PIC18F26K22
IR-LED's

➡ beginners
gevorderden
experts

Ongeveer 1 uur

Standaard soldeeruitrusting

Ongeveer € 40

de interne EEPROM van de microcontroller kunnen maximaal 15 metingen worden opgeslagen. Elke meting bevat de tijdsperiode, de totale afstand en de topsnelheid. K3 is pincompatibel met een Microchip Pickit-programmeerinterface en kan worden gebruikt om de microcontroller te programmeren. Met drukknop S1 resetten we de PIC.

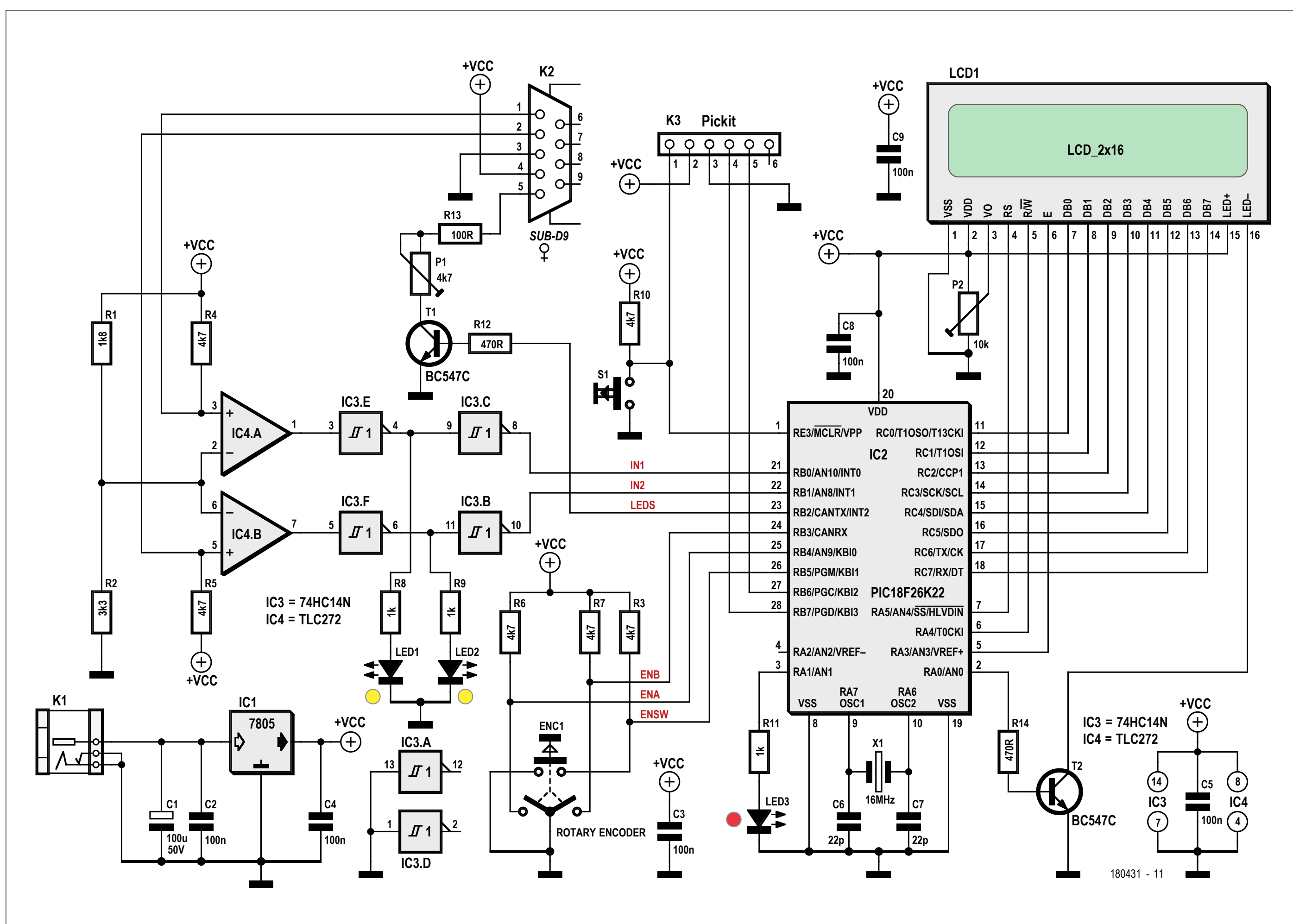
Via een standaard 2×16-karakter-LCD laten we de instellingen en gegevens weergeven. P2 regelt het contrast van het display en de microcontroller schakelt de achtergrondverlichting via transistor T2 in- en uit. ENC1 is een draaiknop met geïntegreerde drukknop voor het wijzigen van display en instellingen.

De IR-LED's zijn parallel aangesloten tussen pen 4 en 5 van K2 (VCC op de anodes en de stuurspanning op kathodes). Met P1 bepalen we de intensiteit van de leds. Transistor T1 geleidt de stroom en ontlast de PIC.

Voor het prototype in het Elektor Lab hebben we twee TCRT5000-IR-reflectiesensoren gebruikt om het draaien van het wiel te registreren, maar vrijwel elke andere combinatie van IR-led en detector zal wel functioneren. Het signaal dat de PIC genereert, is een gemoduleerde blokgolf van 38 kHz, dus een TSOP4838-afstandsbedieningssensor (of elk ander 38-kHz-type) in combinatie met een IR-LED (bijvoorbeeld een CQY99) zal de truc ook doen. Deze laatste oplossing is iets volumineuzer en moeilijker wat betreft de mechanische constructie, maar de modulatie maakt in ieder geval dat de sensoren minder gevoelig zijn voor andere (IR-)lichtbronnen.

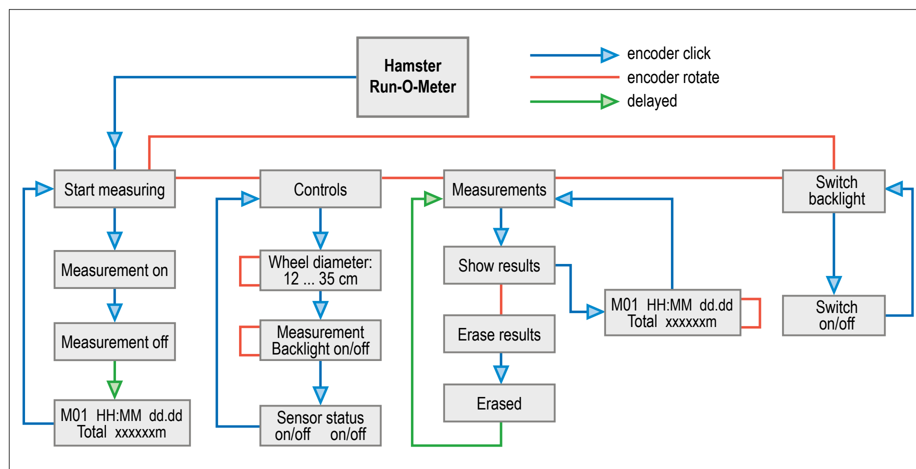
Een kanttekening: de opamp (IC3) en Schmitt-trigger-poorten (IC2) zijn niet per se nodig als u een afstandsbedieningssensor gebruikt, maar ze doen ook geen kwaad.

De emitters van de TCRT5000s zijn aangesloten op GND (K2 pen 3). De collector van de ene sensor is aangesloten op pen 1, de andere op pen 2 van K2. IC3 en IC4 filteren de uitgangssignalen van



Figuur 1. De PIC-microcontroller stuurt het LCD aan en houdt de metingen bij. Bedienen gebeurt met ENC1.

de detectoren en schonen ze op voor de microcontroller.
LED3 knippert bij het inschakelen en schakelt in en uit na elke omwenteling van het wiel. De twee andere leds (LED1 en LED2) geven de status van de sensoren aan. Ze lichten op wanneer er weerkaatst licht wordt ontvangen. Deze indicatie loopt parallel aan de optie ‘Sensorstatus’ op het LCD (daarover later meer), maar de LED’s reageren sneller dan het relatief langzame LCD.
De voeding stabiliseren we met de bekende 7805 in TO-220-behuizing. Zodoende kunnen we een adapter vanaf 8 V gebruiken. Om de warmteontwikkeling in de 7805 binnen de perken te houden, kunt u het beste geen voedingsspanning boven de 15 V aanbieden.



Figuur 2. De menustructuur laat duidelijk zien hoe de Hamster Run-O-Meter bediend kan worden.

Besturing

De menustructuur van onze Hamster Run-O-Meter is weergegeven in **figuur 2**. Met een combinatie van draaien (rode lijnen) en drukken (blauwe lijnen) op de optische encoder bedient u de schakeling. De groene lijn is een automatische (vertraagde) overgang.

Als het LCD is ingeschakeld, geeft de schakeling “Hamster Run-O-Meter” weer. Na een druk op de knop van de encoder wordt het eerste menu-item weergegeven: “Start measuring”. U kunt de meting starten door nogmaals op de knop te drukken of u kunt aan de encoder draaien om door de andere hoofdmenuonderdelen te bladeren.
In het menu “Controls” bevinden zich twee instellingen en één testmogelijkheid. Via de eerste instelling kan de diameter van het hamsterwiel ingesteld worden tussen 12 en 35 cm. Een druk op de knop bevestigt de instelling en slaat de waarde op in het EEPROM. De tweede instelling is de selectie “Measurement Backlight on/off”. Omdat hamsters meestal liever ’s nachts rennen, kan het vervelend zijn als de achtergrondverlichting van het LCD de hele nacht aan staat als de meting loopt. Met deze optie kan de achtergrondverlichting automatisch uitgeschakeld worden wanneer een meting begint en weer worden ingeschakeld wanneer deze wordt gestopt. Door aan de encoder te draaien kunt u “on” of “off” selecteren. Deze instelling wordt opgeslagen in EEPROM wanneer u de encoderknop in drukt. A propos: in de menu-optie “Switch backlight” kan de achtergrondverlichting volledig uitgeschakeld worden, alleen is het blauwe LCD dat wij hebben gebruikt dan wel heel erg lastig af te lezen.
Vervolgens toont het scherm “Sensor status” op de eerste, en twee keer “on”/”off” op de tweede regel (één aanduiding per sensor). Wanneer een sensor geen gereflecteerd licht ontvangt, wordt “off” weergegeven, wanneer hij wel licht ontvangt (de strook op het wiel passeert de sensor), toont het de tekst “on”. Op deze manier kunt u door te draaien aan het wiel vaststellen of de sensor zijn werk goed doet.

Metingen

De volgende optie van het hoofdmenu is “Measurements”. Na een druk op de knop toont een draai aan de encoder “Show results” of “Erase results” op het LCD. Als u nogmaals op de encoder drukt terwijl “Show results” op het LCD staat, kunt u elk van de maximaal 15 opgeslagen metingen bekijken. De volgende gegevens staan op het LCD:

- het nummer van de meting, “Mx” (x loopt van 1 tot 15)
- de duur van de meting in hh:mm-formaat (minimaal 00:01 en maximaal 23:59)
- de maximumsnelheid in xx.xx km/u (snelheden < 0,50 km/u worden niet opgeslagen)
- de totale afgelegde afstand in meters (“Total xxxxxx m”).

Door aan de encoder te draaien springt het LCD door de 15 metingen heen, door erop te drukken keert u terug naar het menu Measurements.
Als u op de knop drukt wanneer “Erase results” wordt weergegeven, worden alle 15 metingen gewist.
De laatste optie in het hoofdmenu is “Switch backlight” waarmee u, zoals de naam al doet vermoeden, de achtergrondverlichting in- en uit kunt schakelen.

Opbouwen van de PCB

Omdat alle onderdelen *Through hole*-typen zijn, is solderen niet al te moeilijk (zie **figuur 3**). Begin met het solderen van de weerstanden, dan de IC-voetjes, condensatoren, enz. Steek de IC’s in de voetjes. K3, de aansluiting voor de programmeerinterface, is alleen nodig als u geen voorgeprogrammeerde PIC hebt of als u met een zelfgemaakte firmwareversie wilt experimenteren. Het kan handiger zijn om deze header aan de onderzijde van de PCB te solderen.
Voor het LCD soldeert u de 16-polige socket aan de componentzijde van de printplaat, de 16-polige header komt aan de achterzijde van het display. Bevestig de vier 10mm-afstandsbusjes op de printplaat, steek het display voorzichtig in de contactstrip en kijk of het LCD mooi op de afstandsbusjes ligt (waarschijnlijk niet). Gebruik, om te voorkomen dat u het LCD beschadigt, onderlegringetjes om het LCD goed te laten aansluiten op de afstandsbusjes vóórdat u het op de busjes vast schroeft.

Mechanische installatie

Kleef een klein strookje (groot genoeg om de IR-LED-bundel naar de detector te reflecteren) aluminiumfolie of een ander reflecterend materiaal op de buitenzijde van het wiel, gericht naar de buitenkant van de kooi, zodat de sensoren net bui-



ONDERDELENLIJST

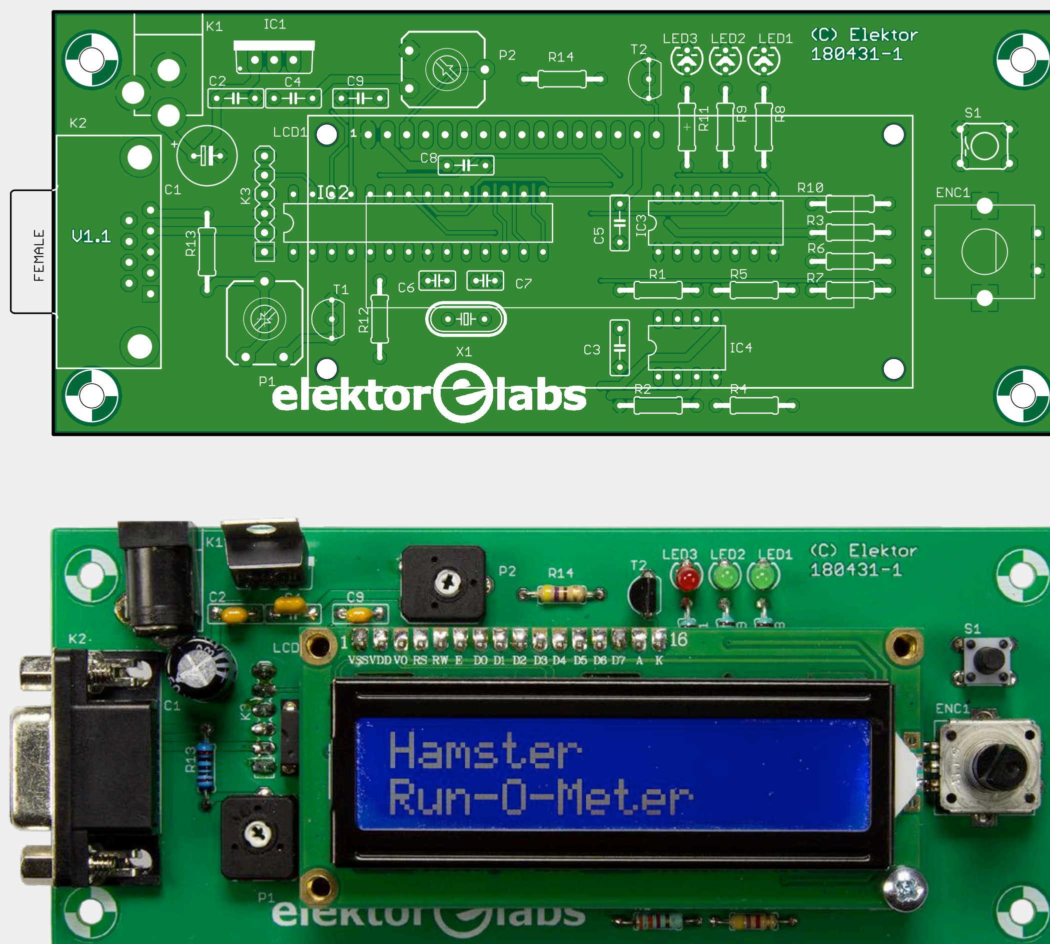
Weerstanden:
R1 = 1k8
R2 = 3k3
R3,R4,R5,R6,R7,R10 = 4k7
R8,R9,R11 = 1 k
R12,R14 = 470 Ω
R13 = 100 Ω
P1 = 4k7 instelpot, horizontaal
P2 = 10 k instelpot, horizontaal

Condensatoren:
C1 = 100 µF/50 V, steek 3,5 mm
C2,C3,C4,C5,C8,C9 = 100 nF/50 V, X7R, steek 5,08 mm
C6,C7 = 22 pF/50 V, C0G/NP0, steek 2,5 mm

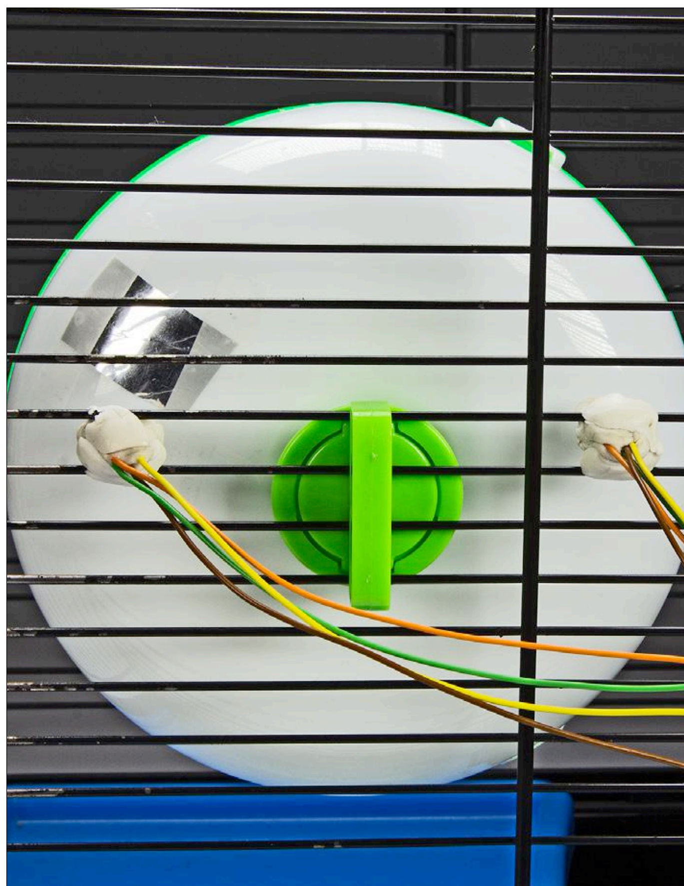
Halfgeleiders:
IC1 = MC7805, 5 V, 1 A
IC2 = PIC18F26K22 8-bit microcontroller, geprogrammeerd, EPS 180341-41
IC3 = 74HCT14 Schmitt-trigger inverter
IC4 = TLC272CP, dubbele JFET-opamp.
IC5,IC6 * = TCRT5000, IR reflectiesensor (NIET op PCB)
IC5,IC6 * = TSOP4838, ontvanger voor afstandsbediening 38kHz (NIET op PCB)
LED1,LED2 = LED, geel, 3 mm
LED3 = LED, rood, 3 mm
LED4,LED5* = IR LED TSUS5400 (niet op PCB)
T1,T2 = BC547C, 45 V, 100 mA, 500 mW, h_{fe}=400

Diversen:
LCD1 = 2x16-karakter LCD met achtergrondverlichting, (zie 'In de store')
LCD1' = 1x16-weg socket, verticaal
LCD" = 1x16-pens header, verticaal.
S1 = drukschakelaar, 24 V, 50 mA, 6x6 mm
ENC1 = EC12E2424407, incrementele draai-encoder met drukschakelaar, 12 mm, verticaal
X1 = Kristal 16 MHz
K1 = voedingsconnector, 1,95 mm pen
K2 = 9-pens sub-D connector voor printmontage, vrouwelijk
K3 * = 6-pens header, steek 2,5 mm
IC-voet, DIP-28, smal (IC2)
IC-voet, DIP-14 (IC3)
IC-voet, DIP-8 (IC4)
9-pens sub-D connector, mannelijk, kabelbevestiging
Afstandsbus met schroef M3 x 10 mm
Schroef M3 x 5 mm
Veerring M3
Moer M3
Print 180431-1 V1.1

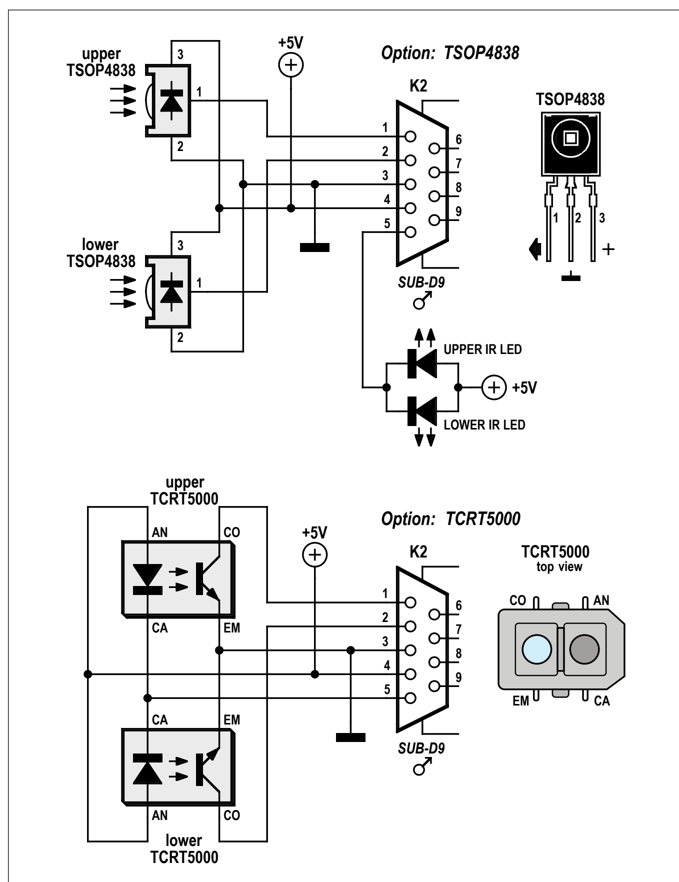
* = zie tekst



Figuur 3. Ons prototype geeft een goed idee van hoe de schakeling opgebouwd dient te worden.



Figuur 4. De sensoren en reflecterende strips zijn het meest effectief wanneer ze op deze manier bij respectievelijk op het wiel geplaatst worden.



Figuur 5. Afhankelijk van het gekozen sensortype kiest u de wijze van bedraden.

ten de kooi geplaatst kunnen worden (knaagdieren houden nu eenmaal van knagen en kabeltjes zijn daar niet tegen bestand). Plaats de sensoren bij voorkeur op (ongeveer) 180 graden ten opzichte van elkaar en het wiel, d.w.z. een halve omwenteling van het wiel (zie **figuur 4**). Afhankelijk van het type sensor dat u wilt gebruiken, zijn er twee manieren om de sensoren te monteren. Bij de eerste methode bestaat elke 'sensor' uit een IR-led (bijvoorbeeld een TSUS5400 of een CQY99, bij voorkeur met reflector) en een TSOP4838 IR-ontvanger die naast elkaar op een klein stukje gaatjesprint zijn gemonteerd. Buig ze lichtjes naar elkaar toe, zodat de detector de weerkaatste IR-bundel van de zender goed kan 'zien'. U zult zelf een soort mechanische constructie buiten de kooi moeten bouwen om deze printen/sensoren te monteren. Uiteraard mogen de IR-bundels niet gereflecteerd worden door de stangen van de kooi. Een kleine tip: houd er rekening mee dat een TSOP4838 zeer gevoelig is voor IR-afstandsbedieningen, dus begin niet te zappen tijdens metingen.

Met methode twee is een kleinere opstelling mogelijk die met een beetje vakmanschap aan de spijlen van de kooi bevestigd kan worden. In dit geval worden twee TCRT5000-combisensoren gebruikt. Deze hebben de collector van een fototransistor als uitgang. Daarom hebben we de opamps (IC4) en Schmitt-trigger inverters (IC3) geïmplementeerd, die de signalen op deze uitgangen opwaarderen naar schone digitale signalen.

Onze ervaring is dat de tweede methode (met de TCRT5000's) minder betrouwbaar is dan de methode met de TSOP4838's. **Figuur 5** toont het bedradingsschema van beide methoden.

Tot slot

Bij beide instellingen is enig experimenteren niet uit te sluiten. Mechanisch is door te 'spelen' met de locatie en hoek van de sensoren (en vooral de afstand tot het wiel) veel bij te stellen. Elektrisch kan bijvoorbeeld de stroom door de LED's met potentiometer P1 gevarieerd worden voor betere resultaten.

De software voor de microcontroller is overigens zoals altijd gratis te downloaden van de webpagina bij dit project [1].

(180431-01)

IN DE STORE

→ Print v1.1
www.elektor.nl/180431-1

→ Geprogrammeerde PIC18F26K22-I/SP-microcontroller
www.elektor.nl/180431-41

→ 2x16-karakter LCD met achtergrondverlichting
www.elektor.nl/120061-74

Weblinks

[1] Webpagina bij dit artikel: www.elektormagazine.nl/180431-01

eLABS PROJECT

Wekker met drievoudige weergave — Update

firmware versie 2.6.1

Mathias Claussen en **Thijs Beckers** (Elektor Labs)

Voor de *Wekker met drievoudige weergave* (Elektor mei/juni 2018, [1]) werd de wens uitgesproken meer dan één alarm in te kunnen stellen. Wie wil er immers vroeg gewekt worden op zijn vrije dag als je kunt uitslapen? Dus moesten er 'even' een paar alarmen aan de software toegevoegd worden. Wie wel eens software heeft ontwikkeld, weet dat dergelijke aannames — vooral bij bestaande code — de praktijk niet altijd even correct benaderen. "Iemand is er al aan begonnen. Het zag er al bijna klaar uit en zou niet zo veel werk meer moeten zijn". Ja ja...

Wat is er aangepast?

Negen alarmen in totaal krijgt de wekker met deze update. Waarom negen? Nou, dat past mooi in het display en het is meer dan genoeg om de week door te komen. Daarnaast is het automatisch dimmen van het display te deactiveren en zijn er drie helderheidsniveaus in te stellen.

Er zijn nu twee menu's voor de bediening: één voor het instellen van de tijd en de achtergrondverlichting en één voor het instellen van de alarmen (zie **figuur 1** en **2**). Met de S1-toets roep je de instellingen voor het instellen van de tijd op. Hier hebben we het item "Backlight" toegevoegd, waar je kunt kiezen of de wekker automatisch moet dimmen of één niveau aanhoudt. Ook kun je hier de LDR (die het omgevingslicht detecteert) kalibreren. Met de S2-toets kom je in het instelmenu voor de alarmen.

En nu we toch bezig waren, hebben we hier en daar wat optimalisaties aangebracht. Zo zijn de beeldopbouw, het RAM-gebruik en de rekentijd geoptimaliseerd. Wanneer u alleen geïnteresseerd was in wat er geactualiseerd is, bent u nu klaar met het artikel, maar wanneer u meer wilt weten over wát we precies gedaan hebben, leest u vooral verder!

Achtergrondverlichting en knoppen

Voor de achtergrondverlichting gebruiken we Timer1 zoals in het in mei/juni-nummer gepubliceerde artikel is beschreven, maar om de aansturing te vereenvoudigen, schrijven we in de huidige schets niet direct in de registers van Timer1, maar gebruiken we de TimerOne-bibliotheek. Hiermee wordt de code voor Timer1 en het genereren van de PWM voor de achtergrondverlichting een stuk gemakkelijker:

```
Timer1.initialize (1000); //
   initialiseert Timer1 en stelt
   hem in op 1000 Hz
Timer1.pwm(9, 512); // stel pwm in
   op pin 9, 50% duty-cycle
Timer1.attachInterrupt(Callback);
   // koppelt Callback() als een
   timer overflow-interrupt
```

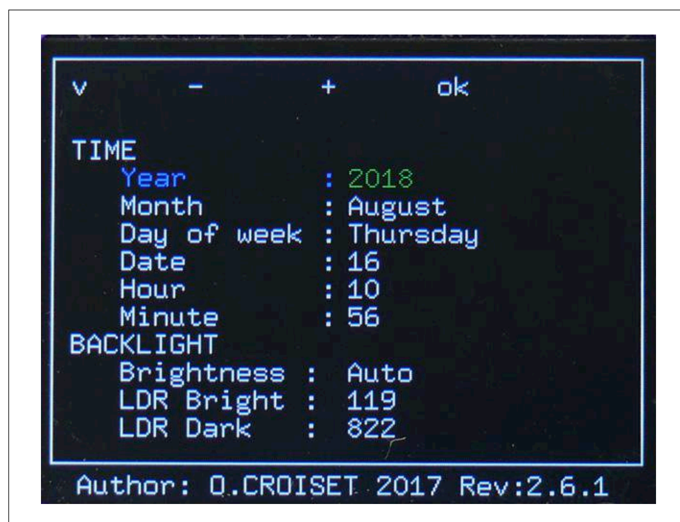
Met `Timer1.initialize` specificeren we de periode in microseconden, de bibliotheek zorgt voor de conversie naar de juiste registerwaarden voor de AVR. De duty-cycle van het PWM-signaal loopt van 0 tot 1023 en 50% komt overeen met de waarde 512. De laatste regel is voor de overflow-interrupt: wanneer Timer1 het einde van een periode bereikt, wordt een interrupt gegenereerd. Hier wordt de functie `Callback()` aangeroepen. Dit gebeurt hier dus 1000 maal per seconde en we sturen de PWM-generator dus met 1000 Hz aan, waardoor het display voor het menselijk oog niet meer flikkert.

Voor het uitlezen van de drukknoppen zou je misschien in eerste instantie overwegen om Timer2 te gebruiken, maar de functie `tone()` (regelt het zoemen van de buzzer) gebruikt Timer2 al om het gewenste geluid te genereren. En Timer0 wordt gebruikt in de Delay-functies. Rest ons Timer1. Omdat we de knoppen ongeveer 100 maal per seconde willen uitlezen, moeten we er in `Callback` voor zorgen dat de uitleesfunctie elke 10e keer wordt aangeroepen.

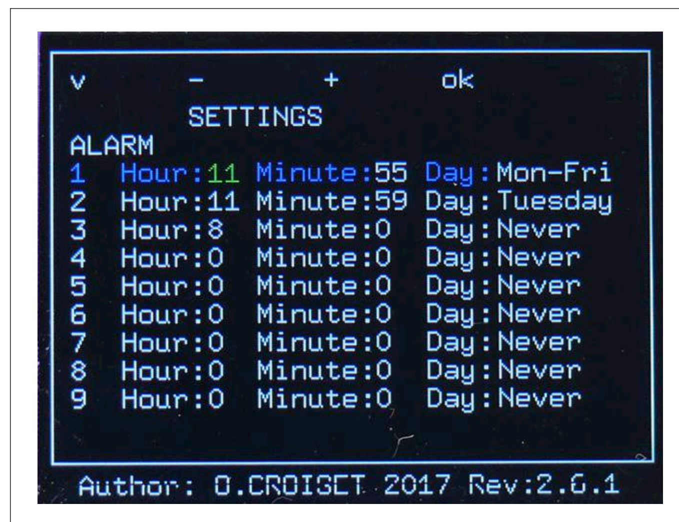
Daarnaast hebben we de zoemer die afgaat op het ritme van een seconde aan, een seconde uit. Om stabiel elke seconde te wisselen, gebruiken we de Callback-procedure in de Timer1-interrupt en zorgen we ervoor dat de functie voor het inschakelen van de buzzer elke seconde wordt aangeroepen.

Wanneer moet het alarm af gaan?

Met de S2-toets open je het alarmmenu (alleen als er géén alarm af gaat; als er



Figuur 1. De klok is met de nieuwe software in te stellen via twee menu's. Hier zien we het menu voor het instellen van de tijd en de achtergrondverlichting...



Figuur 2. ... en hier zien we het menu voor het instellen van de negen alarmtijden.

een alarm afgaat, heeft de S2-knop de functie van de sluimerknop en gaat het alarm na 5 minuten weer af). In dit menu stel je de tijd en de dag van de week waarop een alarm af gaat in voor de individuele alarmen. De instelling "Mon-Fri" zorgt ervoor dat het alarm van maandag tot vrijdag actief is en "Never" deactiveert het alarm.

Om te bepalen wanneer de wekker af moet gaan, berekent hij elke minuut de resterende tijd tot het volgende alarm. Het alarm met de eerstvolgende tijd wordt als volgende alarm weergegeven op het scherm. Klinkt eenvoudig, maar heeft een paar bijzonderheden.

Voor het volgende voorbeeld gaan we er van uit dat het vrijdag 11:56 is en we de volgende alarmen hebben ingesteld:

Alarm	Uur	Minuut	Dag
1	11	55	ma-vr
2	11	59	di
3	08	00	nooit

Alarm 1 gaat van maandag tot en met vrijdag elke dag af. We moeten dus weten welke dag het is. Als het zaterdag of zondag is, gaan we ervan uit dat alarm 1 bedoeld is voor de maandag erna. Als het niet een van deze twee dagen is, nemen we aan dat het alarm op dezelfde dag afgaat.

Nu is het vrijdag 11:56 uur. Bovenstaande logica vertelt ons dat het alarm op dezelfde dag moet afgaan (sterker nog, dat moet hij al gedaan hebben). De software telt de dagen af (in dit geval dus 0), dan de uren (weer 0) en dan de minuten (hier –1). Deze –1 zegt de software dat dit alarm al is afgegaan en dat we enkele correcties moeten aanbrengen.

We trekken 60 minuten (1 uur) af van de uren en tellen ze op bij onze minuten. Dat geeft ons 59 minuten. Er waren nog 0 uren, dus komen we op –1 uur uit. Dus moeten we deze corrigeren en trekken we 24 uur (1 dag) af van de dagen en tellen ze op bij de uren. Resultaat: 23 uur. Nu zijn de uren weer positief, maar de dagen niet meer (–1 dag). Hier wordt het wat lastig. Het alarm is geldig van maandag tot vrijdag. Dus als het vandaag maandag, dinsdag, woensdag of donderdag is, kunnen we de –1 gewoon vervangen door 0 alles gaat goed. Maar het is vrijdag en omdat zaterdag het alarm niet mag af gaan, moeten we er nog drie dagen aan toevoegen (zaterdag en zondag + één geleend). Zo komen we op 2 dagen, 23 uur en 59 minuten voor het alarm weer afgaat. Om de AVR er beter mee te kunnen laten rekenen, tellen we alles in minuten. Dus voor dit alarm komen we op 4319 minuten.

Bij alarm 2 is het eenvoudiger omdat dit slechts op één dag van de week actief is. Eerst berekenen we de hele dagen weer tot de alarmtijd. We beginnen op maandag bij nul. Het alarm staat op dinsdag (1) en het is nu vrijdag (4), levert 1 – 4 = –3 dagen. In dit geval tellen we er, omdat het alarm slechts 1x per week gaat, 7 dagen bij op en komen we op 4 dagen uit. Het alarm staat op 11 uur; het is nu 11 uur, dus er is nog 11–11 = 0 uur te gaan. Het alarm staat op minuut 59, dat is nog 3 hele minuten. Resultaat: 4 dagen, 0 uur en 3 minuten te gaan, oftewel 5763 minuten.

Voor alarm 3 is de berekening zeer eenvoudig, want dit staat uit en mag nooit af gaan. We nemen hiervoor 65535 minuten, wat meer dan een week is en dus ongeldig.

Na al dit gereken hebben we het volgende:

- Alarm 1: 4319 minuten
- Alarm 2: 5763 minuten
- Alarm 3: 65535 minuten

Alarm 1 moet dus als eerstvolgende af gaan.

Optimalisatie

De Arduino-code heeft al ongeveer 30 kB van de beschikbare 32 kB aan flashgeheugen nodig en neemt 1192 bytes RAM van de maximaal 2000 bytes in. Een kritische blik op de programma-omvang kan dus geen kwaad.

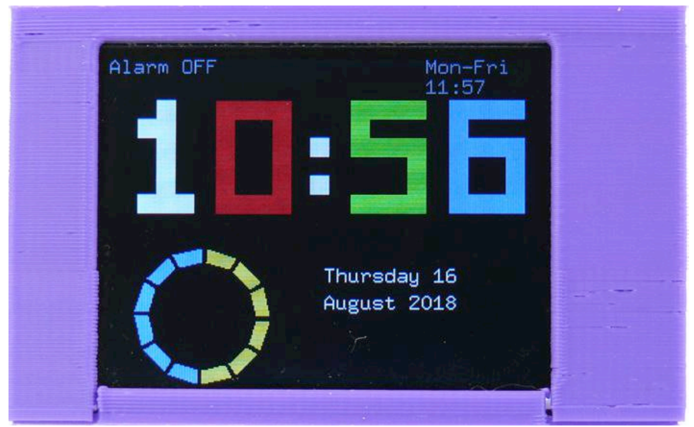
Bij het bekijken van de code voor de Arduino-sketch is het eerste aandachtspunt het type variabelen. Daar vinden we o.a.

```
double-x1Trait; // internal circle
   Cinqmin x
double-y1Trait; // internal circle
   Cinqmin y

float x1precH = 120;
float x2precH = 120;
```



Figuur 3. Het tekenen van de klok lijkt waarschijnlijk eenvoudiger dan het voor de AVR is.



Figuur 4. De secondencirkel vraagt ook de nodige berekeningen.

```
int myYear;
int myMonth;
```

Hoe gemakkelijk het ook is om in de code te werken met float, double en int, voor de AVR is het niet de makkelijkste kost. Ook gebruiken we soms meer RAM dan nodig. Bij doubles en floats is er bij de AVR-compilers iets bijzonders aan de hand. Terwijl moderne architecturen bij een double met 64 bits en bij een float met 32 bits rekenen, is dit bij de AVR-compiler (hier AVRGCC) gelijk getrokken. Dit is een beetje bedrieglijk en zorgt uiteindelijk niet voor een hogere nauwkeurigheid.
Ook valt op dat veel variabelen met int worden gedeclareerd. Een int betekent dat we twee bytes gebruiken en dat we aantallen van -32.768 tot 32.767 kunnen opslaan. Met deze overweging zien we of we de code snel kunnen optimaliseren zodat we minder rekentijd nodig hebben en zowel RAM- als flash-gebruik kunnen minimaliseren.
Bij het ontwikkelen van code voor microcontrollers probeer je de code zo te ontwerpen dat alle bewerkingen uitgevoerd kunnen worden met de oorspronkelijke breedte van de registers in de CPU. Bij de AVR zijn deze 8 bits breed (een byte) en kunnen we twee bits optellen, aftrekken of (alleen ATmega) vermenigvuldigen binnen één kloktik. Maar als we waarden hebben die meer dan één byte groot zijn, duurt het langer. Dan moet de compiler de berekeningen in 8-bit-bewerkingen opsplitsen. Hoewel dit soort bewerkingen al de nodige extra inspanning vergen, vallen berekeningen met floats of double voor de AVR in een heel andere moeilijkheidsgraad. Als dan ook nog functies als sinus of cosinus gevraagd worden, spreken we al snel over enkele *duizenden* kloktikken.

Rekenen met goniometrische functies beperken

Op dit moment heeft de software aardig wat rekentijd nodig om het scherm op te bouwen. Een van de oorzaken hiervoor is de ucglib-library die het zeer eenvoudig maakt om het display aan te sturen, maar die erg veel rekencapaciteit kost. Daarnaast vergen de trigonometrische functies die gebruikt worden voor het bepalen waar de wijzers getekend moeten worden en waar de seconde-punten moeten staan (zie **figuur 3**) ook de nodige `sin()` en `cos()`-berekeningen. Alleen al voor de hoekpunten van de twee wijzers zijn 8 coördinaten nodig (elke hoek één) die op denkbeeldige cirkels geplaatst moeten worden — waarvoor met `sin()` en `cos()` gerekend wordt. Voor een `sin()`-berekening heeft de AVR ongeveer 1650 cycli nodig, voor een `cos()` evenveel.
Laten we eens kijken naar de functie SecondeSecteur, die de 'secondencirkel' linksonder tekent (zie **figuur 4**). Hiermee wordt elke 5 s een nieuw segment getekend. Hiervoor zijn per segment `sin()` en `cos()` elk vier keer nodig. In de broncode zien we:

```
x_ext_Sec = ext_radius *
   cos(angleSec); x_ext_Sec = x_
   ext_Sec + xcenter ;
y_ext_Sec = ext_radius *
   sin(angleSec); y_ext_Sec = y_
   ext_Sec + ycenter ;
x_int_Sec = int_radius *
   cos(angleSec); x_int_Sec = x_
   int_Sec + xcenter ;
y_int_Sec = int_radius *
   sin(angleSec); y_int_Sec = y_
   int_Sec + ycenter ;
x_ext_SecPrec = ext_radius *
   cos(angleSecPrec); x_ext_
   SecPrec = x_ext_SecPrec +
   xcenter ;
y_ext_SecPrec = ext_radius *
   sin(angleSecPrec); y_ext_
   SecPrec = y_ext_SecPrec +
   ycenter ;
x_int_SecPrec = int_radius *
   cos(angleSecPrec); x_int_
   SecPrec = x_int_SecPrec +
   xcenter ;
y_int_SecPrec = int_radius *
   sin(angleSecPrec); y_int_
   SecPrec = y_int_SecPrec +
   ycenter ;
```

Alle variabelen zijn integers. Elke vermenigvuldiging van twee integers heeft 20 cycli nodig. Dus komen we zonder alle andere berekeningen uit op (1650×8) + (20×8) = 13360 cycli. Bij 12 MHz is dat 1,11 milliseconden. En er zijn 12 segmenten die elke 5 s berekend moeten worden. Om de microcontroller te sparen, kunnen we vooraf de benodigde waarden berekenen en met een tabel werken. Hierdoor kunnen we de tijd verkorten van enkele duizenden cycli tot minder dan 100.
Hoewel dit niet het leeuwendeel van de tijd voor tekenen bespaart, toont het aan dat een beetje nadenken tijd en code kan besparen. Dezelfde aanpak gebruiken we

voor de functie `AiguilleSecondes`.
Op andere plaatsen zou hetzelfde mogelijk zijn voor de minuten- en uurwijzers, maar hier zijn (8×60) + (8×12×60) = 6240 waarden nodig, omdat elke wijzer elke minuut een andere positie heeft. Dat zou wel een erg grote tabel worden.

Voorkom dubbel werk

Functies voor het tekenen van elementen kun je natuurlijk allemaal separaat laten uitvoeren, maar als onderdelen er hetzelfde uitzien, scheelt het een hoop werk als je ze combineert. Dit hebben we gedaan voor de functies SegmentA() t/m SegmentG(). Omdat de basisfunctie hetzelfde is, kunnen we deze samenvoegen tot een segmentfunctie en de parameters laten bepalen in welk segment we welke kleur willen tekenen.
Met bestaande code zijn dergelijke optimalisaties soms moeilijk te maken, omdat het grote veranderingen kan betekenen.
Ook `Cadran2()` hebben we aangepakt. De Multiple Switch-constructie voor de uren en minuten hebben we vervangen door een hulpfunctie. Dit maakt de eigenlijke functie `Cadran2()` compacter.
Naast het puur vermijden van code kan ook de uitvoering, hier het hertekenen van getallen, vermeden worden. Als de uuraanduiding niet verandert, waarom zou de microcontroller hetzelfde nummer dan nogmaals moeten tekenen? Om dit in elke run van de functie te kunnen controleren, moeten we weten welke getallen we hebben getekend. Dit kan met globale variabelen of met statische variabelen binnen de functie. We hebben de tweede optie geïmplementeerd. Als de huidige tijd afwijkt van de weergegeven tijd, kan nu alleen het gewijzigde gedeelte worden bijgewerkt. Dit zorgt voor een veel aangenamere update van de tijd.

Debuggen zonder debugger

Al deze veranderingen moesten we natuurlijk testen. Het eenvoudigst is natuurlijk een debugger. Maar de Arduino IDE schiet hier helaas tekort. Gelukkig kunnen we altijd nog teruggrijpen op de IO-pen met LED en serieweerstand. De LED kan dan op bepaalde punten in de code in- en uitgeschakeld worden. Dit kost slechts één klokcyclus.
Een andere methode maakt gebruik van de USART. Als de TX-pen niet wordt gebruikt, kan direct leesbare tekst worden uitgevoerd via een `Serial.print()`. Dit is op zich wel praktisch, maar met 9600 baud nemen langere tekenreeksen een aanzienlijke rekentijd in beslag en vertragen ze zichtbaar de programmasnelheid van de wekker. Met het 12MHz-kristal kunnen we prima 250.000 baud instellen, dat loopt een stuk beter. Ook de USB naar serieel-omvormers met CH340- of CH341-chip hebben geen probleem met deze baudrate.
Voor het debuggen hebben we een combinatie van USART en IO-pen met logic analyzer gebruikt. Zo konden we naast het veranderen van het signaal op de pen ook het tijdsgedrag meten. De USART helpt dan met een geschikte `printf()` om eventuele fouten op te sporen.
Maar wat als debuggen niet langer nodig is? Elke `printf()` handmatig uit de code verwijderen? Het is makkelijker om hiervoor de preprocessor aan het werk te zetten:

```
#define DEBUGPRINT( X ) Serial.
   print( X )
//#define DEBUGPRINT( X ) do{ }
   while( 1 ==0)
```

Als we de uitvoer willen activeren, gebruiken we de eerste `#define`. We kunnen dan met `DEBUGPRINT (“Test”)` de tekst “Test” laten uitvoeren. Als we de uitvoer willen stoppen, gebruiken we de tweede `#define`. Het do-while-statement zorgt ervoor dat de compiler niets naar de USART uitvoert. Bij de ingestelde optimalisatie wordt het statement zelfs volledig weg-geoptimaliseerd.
Het weg-optimaliseren bemoeilijkt ook het gebruik van bijvoorbeeld een AVR-Dragon. Bij het optimaliseren naar de kleinste omvang hoeft de compiler er alleen maar voor te zorgen dat het resultaat aan het einde van de functies correct is. De manier waarop hij dat doet, mag hij zelf bepalen en dat kan ervoor zorgen dat de door ons bedachte programmavolgorde volledig omgegooid is of zelfs dat complete functies weggelaten zijn in de optimalisatie.

Tot slot

De wekker heeft nu een overvloed aan alarmmogelijkheden en meer bedieningsgemak. De optie om de LDR te kalibreren bleek nodig nadat we vaststelden dat lang niet elke LDR naar wens functioneerde in de schakeling. De spreiding in de productie lag simpelweg veel te hoog. Verder hebben we een tipje van de sluier opgelicht over het programmeren van de wekker, allemaal *ter leering ende vermaeck*. De nieuwe firmware is overigens te downloaden van [2]. ◄

170112-B-01

Weblinks

[1] Webpagina van het oorspronkelijke artikel in Elektor mei/juni 2018: www.elektormagazine.nl/magazine/elektor-201805/41513

[2] Webpagina bij dit artikel: www.elektormagazine.nl/170112-B-01

[3] Labpagina bij dit project: www.elektormagazine.com/labs/3-displays-alarm-clock-with-tft-screen-1

Universal Case System

flexibele oplossing voor elektronica behuizingen

Thijs Beckers

Wie embedded systemen ontwikkelt, kent het vast wel: het moet vooral betrouwbaar en liefst gemakkelijk aan te passen zijn. Dat geldt ook voor de behuizing. En als het even kan, moet het ook nog een beetje toekomst zeker zijn, zodat de productie niet in gevaar komt. Hiervoor vonden we een goede oplossing bij Phoenix Contact.



Wanneer je in een professionele omgeving embedded systemen ontwikkelt, komt de betrouwbaarheid van het product natuurlijk op de eerste plaats. Maar het is zeker niet onbelangrijk om de concurrentie een stap voor te kunnen zijn. En je moet het ontwerp natuurlijk snel kunnen aanpassen aan de project-eisen, wanneer die (weer eens) veranderen. Het elektrisch ontwerp moet u helaas nog zelf doen, maar voor de behuizing is het UCS (Universal Case System) van Phoenix Contact bijzonder interessant.

De UCS-behuizing (zie de illustratie op de rechter bladzijde) bestaat uit twee identieke helften, uitneembare zijpanelen en opvallende hoekstukken. Voor meer inbouwruimte vervang je eenvoudig de zijpanelen en schroeven door een 20 mm hogere versie, aangevuld met extenderstukjes voor de hoeken.

De functionaliteit van de behuizing pas je gemakkelijk aan bij het samenstellen van de case via een online tool (www.phoenixcontact.nl/configurator-housing).

Configuratie-opties

Het UCS-systeem is zo ontworpen dat je het langere zijpaneel ook kan gebruiken als het kortere zijpaneel in de volgende grotere behuizingsvariant. Zo kun je dit eenvoudig te bewerken paneel in verschillende toepassingen gebruiken. PCB’s kun je op twee manieren monteren:

- met een geïntegreerd schroefgat direct aan de gekleurde hoekstukken. Hiermee benut je het maximaal mogelijke printoppervlak.
- met speciaal ontworpen zelfklevende schroefmontagepunten die je zelf op elke plaats in de behuizing kunt plakken. Hiermee kun je elke PCB-vorm monteren die je maar weet te bedenken.

Het is natuurlijk ook mogelijk om meerdere printen tegelijk in de behuizing op te nemen, bijvoorbeeld door in elk van de twee halve schalen een print te monteren en ze daarna samen te voegen tot één geheel.

Snel een geschikte behuizing en online te configureren.

De lange zijde van de behuizing kan ook worden gebruikt als de korte zijde in de behuizing van één maat groter in de serie. Handig wanneer er al uitsparingen in gefreesd zijn maar als blijkt dat de behuizing te klein is.

Behuizing niet hoog genoeg? De hoekstukjes zijn in verschillende hoogtes te krijgen.

Met de zelfklevende schroefuitsparing bepaalt u zelf de positie van de printplaat.

Extra gemak

Voor een aantal bekende embedded platvormen (bijvoorbeeld voor de Raspberry Pi) biedt Phoenix Contact kant-en-klare zijpaneeltjes aan, zodat je het 3D-ontwerpen van een behuizing, toch wel een zwak puntje bij veel elektronici, kunt besparen.

Naast de behuizingen heeft Phoenix Contact ook erg handige printconnectoren. Hiermee is het heel eenvoudig om externe aansluitingen te realiseren voor alle voeding,- data- en signaalverbindingen. Neem hiervoor eens een kijkje in het COMBICON-assortiment.

Kijk zelf

Nieuwsgierig geworden? Bekijk de video's op YouTube en probeer zelf de Configurator uit op de website van Phoenix Contact. ◂

180475-01

Weblink

[1] www.phoenixcontact.nl/embedded



www.phoenixcontact.nl/embedded

Video's:

https://youtu.be/QYKpTPch16c https://youtu.be/qo_3c4AG6Oo https://youtu.be/K3qwO7HX5xM

DDS met Arduino

interessante experimenten met simpele middelen

Roland Stiglmayr

Het principe van Directe Digitale Synthese (DDS) wordt tegenwoordig in vrijwel alle apparatuur voor draadloze communicatie-, hoogfrequent- en audio-meettechniek toegepast. Vanwege de extreem hoge frequentieresolutie, de programmeerbaarheid van de golfvorm en de mogelijkheid om vrijwel zonder vertraging van frequentie te wisselen, vormt DDS een geducht alternatief voor PLL-synthesizers.

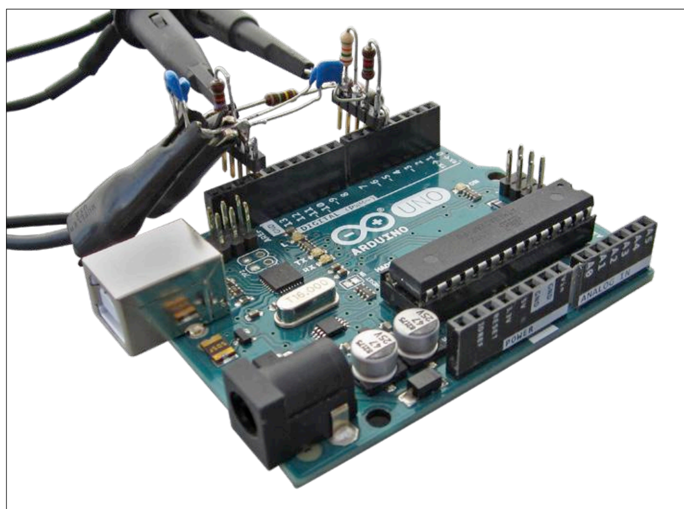
Hoe DDS en DDS-generatoren werken, is zijdelings al diverse malen in Elektor aan de orde gekomen; in de Duitstalige uitgave van Elektor september 2016 [1] is uitgebreid op DDS ingegaan. Veel lezers zijn in elk geval nieuwsgierig geworden; reden genoeg om enkele DDS-experimenten voor te stellen. En dat natuurlijk het liefst op een eenvoudig experimenteer-platform, dat met weinig moeite zelf te bouwen is. Wat ligt er dan meer voor de hand om daarvoor het elektronische manus-je-van-alles te nemen, de Arduino?

En dat is precies wat we hier gaan doen. Wat u ervoor nodig heeft is slechts een Arduino Uno en een eenvoudig laagdoorlaatfilter. Het filter bestaat uit een handjevol onderdelen die ‘zwevend’ worden bedraad en in de uitbreidingsconnector van de Arduino worden geprikt, zoals te zien in **figuur 1**. Wie zich wat meer in deze materie wil verdiepen, doet er goed aan om met gaatjesprint een eenvoudig shield te maken (zie **figuur 2**). Het voornaamste doel van dit project is niet de realisatie van een DDS-generator. Het gaat ons hier vooral om een beter begrip van deze techniek. Daarom helpt het als u de beschikking hebt over een oscilloscoop, waarmee de verbanden zichtbaar kunnen worden gemaakt. In plaats daarvan is het natuurlijk ook wel mogelijk om het uitgangssignaal, een sinus van 1,9 Hz tot 31 kHz, hoorbaar te maken door middel van een laagfrequent-(audio-)versterker, maar dat is toch veel minder inzichtelijk dan wat u op een ’scoop kunt zien.

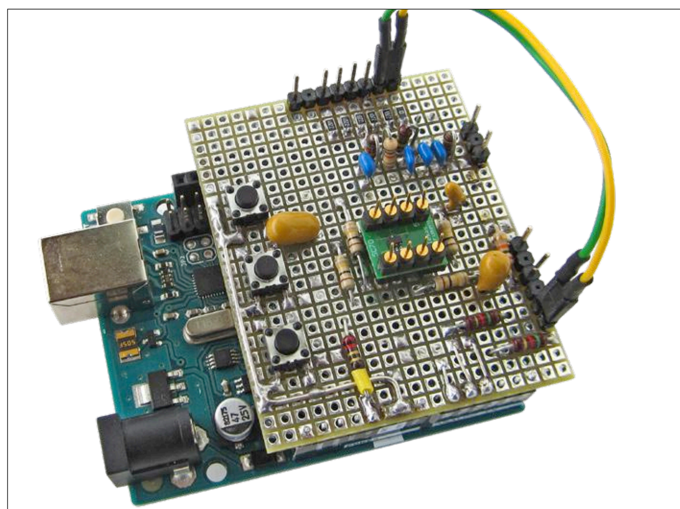
Wat doet dit Arduino-experiment?

Met DDS zijn verscheidene signaalvormen met bijna willekeurige frequenties te maken. Frequentie, fase en amplitude van het uitgangssignaal kunnen vrijwel zonder enige vertraging worden veranderd. Daardoor is DDS welhaast voorbestemd voor het ontwerpen van functiegeneratoren, draaggolf-oscillatoren, sinus/cosinus-generatoren voor I/Q-modulatoren, maar daarnaast kan DDS ook heel goed worden ingezet bij allerlei vormen van digitale modulatie.

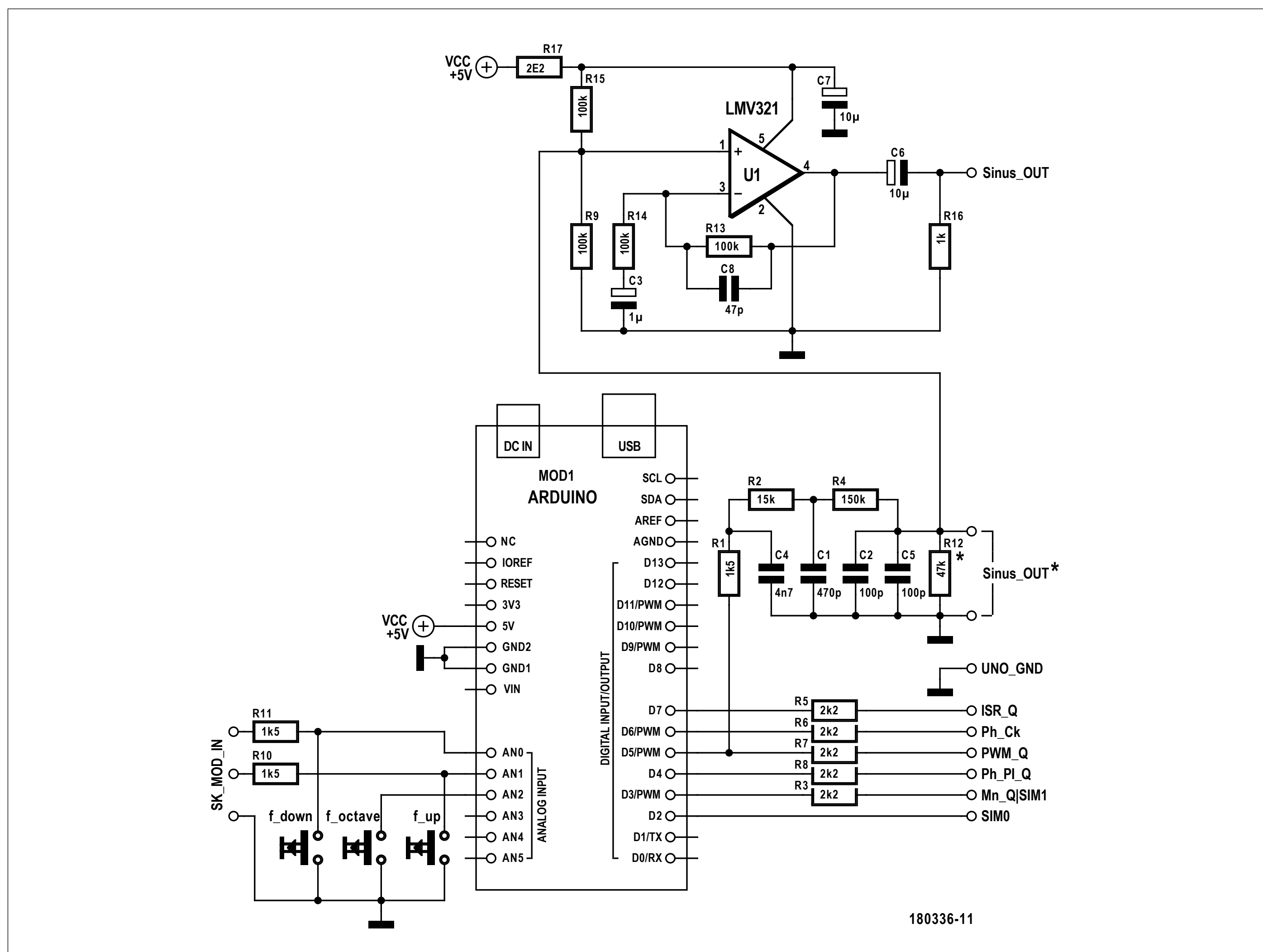
Het Arduino-platform produceert een sinus met een frequentie



Figuur 1. Zwevend bedraad laagdoorlaatfilter. Meetpunt Ph_PI, via de weerstand ontkoppeld, is trigger voor de oscilloscoop.



Figuur 2. Laagdoorlaatfilter, druktoetsen en impedantie-omzetter met een opamp op een zelfgebouwd shield.



Figuur 3. Het schakelingetje voor de Arduino Uno kan zwevend of op gaatjesboard worden gebouwd.

die van 1,9 Hz tot 31 kHz kan worden ingesteld. De frequentie kan in stapjes van 1,9 Hz worden verhoogd en verlaagd, maar ook in stappen van een octaaf. De frequentie wordt ingesteld met drie toetsen. Met wat meer moeite kan ook met digitale modulatie worden geëxperimenteerd.

Als u niet kunt wachten, dan kunt u nu al de sketch "DDS_SIN" downloaden, het laagdoorlaatfilter bouwen volgens het schema in **figuur 3** en beginnen met de eerste metingen. Aangezien echter een diepgaander begrip van de theorie toch wel nodig is bij de meer serieuze probeersels en experimenten, is het waarschijnlijk verstandiger om eerst de hieronder beschreven principes tot u te laten doordringen. Ze zijn eigenlijk best simpel en makkelijk te begrijpen.

Zo werkt DDS

Het centrale element van DDS is de fase-accumulator. Die bevat een binair register dat te allen tijde de momentele fasehoek van het te genereren signaal bevat. Bij elke tik van de DDS-klok telt de accumulator daar een instelbaar fase-increment bij op, dat de frequentie bepaalt. Bij overflow vindt er geen overdracht (*carry*) plaats en ook wordt de accumulator niet weer op nul gezet. Het optellen van fase-incrementen loopt gewoon door vanaf de waarde die in het faseregister stond toen de overflow optrad. Bits hoger dan het meest significante bit van het faseregister hebben we immers niet.

Bij overflow is een periode van 2π voltooid, en dat duurt telkens een periodeduur T_{out}. De periodeduur wordt dus uitsluitend bepaald door de grootte van het fase-increment en de bitbreedte van de fase-accumulator, gedeeld door de klokfrequentie f_{DDS}.

$$(\text{fase-accu}_{max} / \text{fase-increment}) * t_{ck,DDS} = T_{out}$$

Dus de frequentie is

$$\text{fase-increment} / (\text{fase-accu}_{max} * t_{ck,DDS}) = f_{out}$$

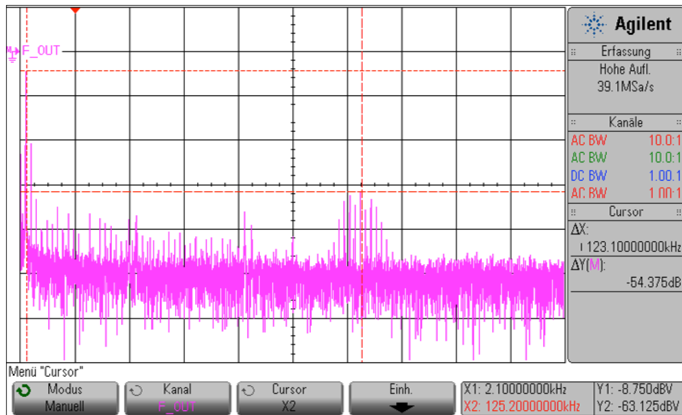
Omdat we werken met gehele getallen is de kleinst mogelijke fasestap gelijk aan één. De laagste frequentie is dan

$$f_{min} = (1 / (\text{fase-accu}_{max} * t_{ck,DDS}))$$

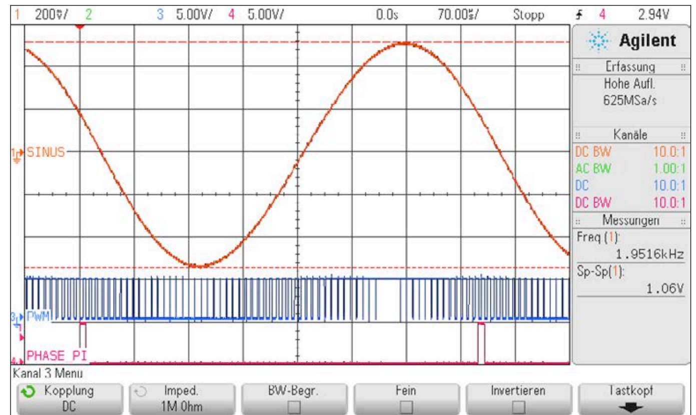
Het fase-increment is een geheel veelvoud van deze kleinste fasestap. Daaruit volgt de uitgangsfrequentie

$$f_{out} = f_{min} * n \quad (\text{met n een geheel getal} \leq (\text{fase-accu}_{max} / 2))$$

f_{min} is daarmee ook de (constante) frequentieresolutie van het systeem. Merk op dat de uitgangsfrequentie lineair afhangt



Figuur 4. Het frequentiespectrum van het uitgangssignaal bij 1953 Hz. f_{mid} = 100 kHz, 20 kHz/div, 20 dB/div.



Figuur 5. Het uitgangssignaal van figuur 4 in het tijddomein.

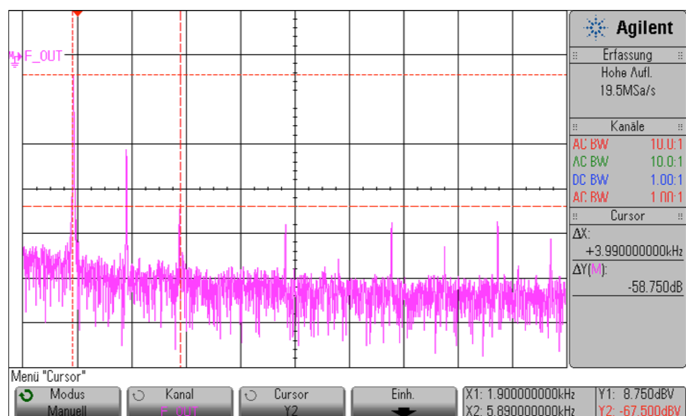
van slechts één factor.
Bij een 2-kHz-signaal bijvoorbeeld zien we het volgende. Een 16 bit brede fase-accumulator loopt over na 216 klokpulsen. Bij een DDS-klok van 125 kHz geeft dat een f_{min} van 125 kHz / 65536 = 1,90734 Hz. Voor 2 kHz hebben we dan een fase-increment nodig van

$$\text{fase-increment}_{2kHz} = 2\text{ kHz} / 1{,}90734\text{ Hz} = 1048{,}58$$

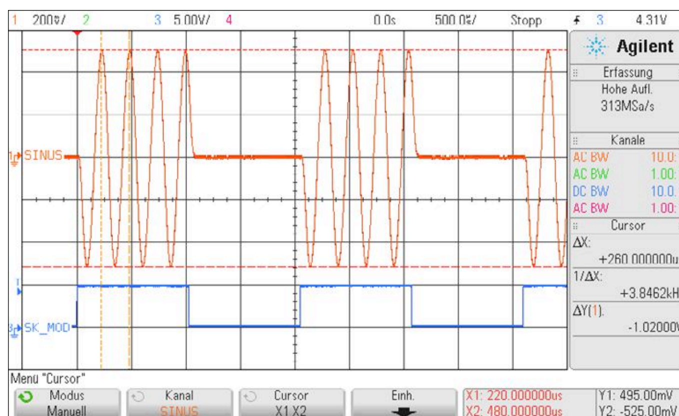
Hier zien we een inherent probleem van DDS: het fase-increment kan slechts een geheel getal zijn, en daardoor kunnen we bijvoorbeeld een frequentie van 2 kHz niet beter dan op ongeveer 1 Hz nauwkeurig instellen. Geïntegreerde DDS-chips lossen dit op met een heel breed register voor de fase-accumulator.
Tot nu toe hebben we het alleen gehad over de fase van het te genereren signaal, maar de amplitudewaarden die horen bij bepaalde fasen zijn natuurlijk ook nodig. Hoe bepalen we die? De oplossing is een opzoektabel (*look-up table* of LUT). Het zou echter enorm veel geheugen kosten als die voor elke fasewaarde een amplitudewaarde zou bevatten. In plaats daarvan bevat hij een waarde voor elke 2n-de waarde. Het aantal waarden wordt dus verkleind door deling van het maximum door 2n, waarbij n zodanig wordt gekozen dat de verschillen tussen de opeenvolgende waarden niet te groot worden. Hier moet dus rekening worden gehouden met de amplituderesolutie. Zo'n opzoektabel kan eenvoudig met Excel worden gemaakt en geoptimaliseerd (zie [1]).
Een voorbeeld ter verduidelijking: met een amplituderesolutie van 7 bit (dus 128 discrete waarden) en een sinus als gewenste golfvorm, hebben we een tabel met tenminste 128 waarden nodig. Daarmee hebben we rond de nuldoorgang, als het signaal de steilste helling heeft, nog voldoende oplossend vermogen. Met een 16 bits fase-accumulator krijgen we in dit voorbeeld een deelfactor van 29.
De aldus gereduceerde waarde van de fase-accumulator dient als pointer naar de bijbehorende regel in de opzoektabel. De uitgelezen amplitudewaarde dient dan als invoer voor een D/A-converter, die dan eindelijk de uitgangsspanning genereert. Deze D/A-converters worden meestal uitgevoerd als R/2R-netwerken. Als de hardware zo simpel mogelijk moet zijn, dan kan hier ook PWM worden gebruikt. Daarna moeten we het signaal nog door een laagdoorlaatfilter sturen om het te ontdoen van resten van de klokfrequentie, hogere harmonischen en mengproducten.

Met software naar de praktijk

Tot zover de (grauwe) theorie. Nu gaat het erom dat we dit alles met simpele software op de Arduino Uno in praktijk brengen. Als u dat nog niet gedaan had, is het nu een goed moment om de sketch "DDS_SIN" [1] in uw Uno te laden.
Net als bij elk ander project is de eerste vraag welke functieblokken we nodig hebben en hoe we die kunnen realiseren. We hebben uiteraard een kloksignaal nodig dat het tempo aangeeft waarmee de fase-incrementen in de fase-accumulator worden opgeteld. Daartoe gebruiken we Timer0, die cyclisch een interrupt-service-routine (ISR) aanroept. Hierin wordt de optelling uitgevoerd, de pointer naar de LUT berekend, de amplitudewaarde uit de LUT uitgelezen en naar de D/A-converter gestuurd. De LUT is een byte-array-variabele in het declaratiegedeelte. De D/A-converter is uitgevoerd als PWM.
En nu zou Kees klaar zijn geweest, ware het niet dat we te maken hebben met de nogal beperkte verwerkingssnelheid van de Uno. Ons doel is een bruikbaar uitgangssignaal van minstens 12 kHz. Met minimaal acht amplitudewaarden per periode hebben we een klok van 96 kHz nodig en liever nog hoger, want hoe hoger de frequentie, des te beter de signaal/ruisafstand – we hebben immers een constante signaalfrequentie.
De teller van Timer0 initialiseren we zodanig dat hij wordt gereset na 128 perioden van de systeemklok. De ISR wordt dan elke 8 µs aangeroepen, dus de DDS-klok tikt op 125 kHz. De PWM moet op dezelfde klok draaien, want hij moet elke opgehaalde amplitudewaarde verwerken. Standaard wordt echter de Arduino-PWM geïnitialiseerd met 256 tijdseenheden, dus op 62,5 kHz. Om dit op te lossen gebruiken we een truc: ook PWM doen we met Timer0. Die telt immers maar tot 128, de gewenste 128 tijdseenheden. De resolutie van de PWM is dan wel beperkt tot 7 bit. Om de uitvoering van het programma zo min mogelijk te vertragen, gebruiken we slechts een paar commando's uit de Arduino-bibliotheek.
Om beter te kunnen interpreteren wat we straks op een oscilloscoop zien bij het doorlopen van het programma, zorgen we voor een paar statussignalen, die nader toegelicht zijn met commentaar in de code. Weerstanden aan de statussignalen dempen storingen die kunnen optreden als we de meetpunten capacitief belasten. Een bijzonderheid van de DDS is het signaal "Ph_PI", dat kort voor de negatieve nuldoorgang tot stand komt. Bij hogere frequenties zien we niet elke periode een Ph_PI-signaal. Dat komt doordat we met gehele getallen



Figuur 6. Het spectrum van het signaal in figuur 4. f_{mid} = 10 kHz, 2 kHz/div, 20 dB/div.



Figuur 7. Het uitgangssignaal met ASK. f_0 = 3906 Hz, bitrate = 1 Kbit/s.

werken en doordat de fase-accumulator niet bij overflow op nul wordt gereset maar doortelt. Daardoor wordt Ph_PI alleen op lagere frequenties bij elke periode gegenereerd.

Met laagdoorlaat en impedantie-omzetter

Elke DDS-generator heeft een laagdoorlaatfilter aan de uitgang. Zeker wanneer (zoals bij onze experimenten) PWM wordt gebruikt, beïnvloedt een laagdoorlaatfilter de kwaliteit van het uitgangssignaal in belangrijke mate. Bij dit project is de kantelfrequentie van het laagdoorlaatfilter bij 12 kHz gelegd, om een decade afstand van de PWM-klok te houden. Bij een derde-orde-filter is de demping per definitie 60 dB per decade, wat ongeveer de waarde is waarmee de PWM-klok en de mengproducten daaromheen worden gedempt. We hadden kunnen kiezen voor een actief filter met een opamp, maar dat was veel meer werk geweest. Bovendien zouden we dan met ‘kromme’ onderdelenwaarden te maken hebben gekregen.
Makkelijker is een passief filter met RC-netwerkjes. Met drie RC-laagdoorlaatfilters achter elkaar hebben we een derde-orde filter. Dat werkt echter alleen wanneer de drie netwerkjes van elkaar ontkoppeld zijn en elkaar niet wederzijds belasten. Dat bereiken we door de ingangsweerstand van elk volgend RC-netwerkje veel hoger te kiezen dan de uitgangsweerstand van zijn voorganger. De kantelfrequentie van de losse trapjes is zo gekozen dat elk trapje een demping van 1 dB geeft op de kantelfrequentie van het filter als geheel. Drie trapjes achter elkaar geven dus 3 dB demping op die ‘overall’-kantelfrequentie. Passen we de overdrachtsfunctie toe, dan krijgen we

$$f_{grens,RC\text{-}trap} = 1{,}98 * f_{grens,out}$$

Kiezen we de componenten zoals in schema aangegeven, dan ligt de kantelfrequentie van een afzonderlijke trap bij 22,6 kHz. Om de hoge uitgangsweerstand van het filter wat te reduceren is de laatste trap voorzien van een tweede weerstand (R12) naar massa. Die reduceert de amplitude, maar geeft wel het gewenste effect.
Spectrumanalyse aan de uitgang van het laagdoorlaatfilter (**figuur 4**) laat zien hoe effectief de dimensionering uitpakt; **figuur 5** toont hetzelfde signaal in het tijddomein. De 125 kHz van de PWM en de mengproducten rond de PWM-frequentie liggen 54,3 dB onder het uitgangssignaal van 1953 Hz.
Als de DDS-generator gebruikt moet worden voor metingen of aansturing van een LF-versterker, dan is het aan te bevelen om een opamp als impedantie-omzetter aan de uitgang te zetten. R12 wordt dan vervangen door R15 en R9, die het DC-werkpunt op $V_{CC}/2$ leggen. De opamp werkt als DC-gekoppelde niet-inverterende versterker. Het signaal gaat vervolgens via een ontkoppelcondensator naar buiten. Zonder R14 en C3 is de versterking A = +1, met R14 en C3 is A = +2.

Sketches

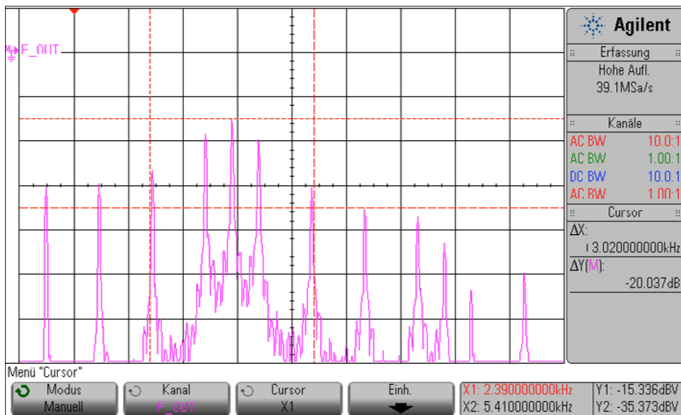
In de download op de projectpagina bij dit artikel [2] vindt u meerdere Arduino-sketches waarin modulatie-methoden geïmplementeerd zijn.

DDS_SIN: sinusgenerator

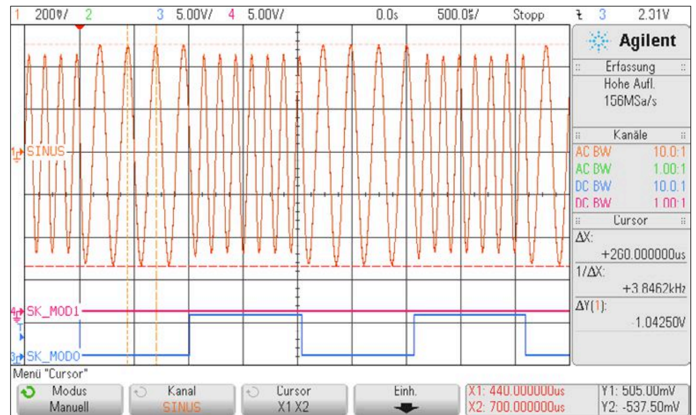
In het hoofdprogramma scannen we de toetsen. Al naar gelang de toets die is ingedrukt wordt het fase-increment verhoogd, verlaagd, of in het geval van octaafstappen verdubbeld. Bij het actualiseren van de frequentie schrijven we de nieuwe waarde via de UART naar de seriële monitor van de IDE. De verschillende stappen in het programma zijn toegelicht in het commentaar in de code.
In **figuur 5** ziet u het uitgangssignaal “SINUS” achter het laagdoorlaatfilter, het signaal Ph_PI waarmee we de oscilloscoop triggeren, en het ingangssignaal van het PWM-filter. De **figuren 4** en **6** tonen het spectrum van figuur 5. In **figuur 6** ziet u de harmonischen en de afstand van de derde harmonische tot de grondtoon, de eerste harmonische.

DDS_ASK: Amplitude Shift Keying

Bij ASK-modulatie wordt de amplitude van de draaggolf verhoogd of verlaagd, al naargelang de logische toestand van de ingang van de modulator. Met DDS is dit heel simpel te realiseren. We definiëren een tabel met amplitudewaarden die bij elke toestand van de ingangsinformatie horen. Alnaargelang die ingangsinformatie wijst de pointer naar de bijbehorende tabelwaarde.
Een speciale vorm van ASK is *ON OFF Keying* (OOK), waarbij de amplitude tussen 0 en 100% wordt geschakeld. In dit experiment passen we deze methode toe. Afhankelijk van de ingangstoestand zet de software het fase-increment ofwel op 3906,25 Hz, ofwel op 0 Hz. De verandering vindt plaats op de nuldoorgang van de sinus. Nadere toelichting vindt u in het commentaar in de sketch.



Figuur 8. Het frequentiespectrum van het ASK-uitgangssignaal van figuur 7. f_{mid} = 5 kHz, 1 kHz/div, 10 dB/div.

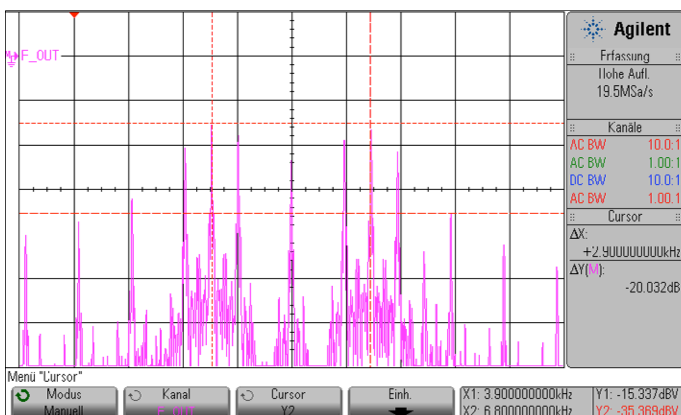


Figuur 9. Het uitgangssignaal met 2-FSK.
f_1 = 3906 Hz, f_2 = 6836 Hz, bitrate = 1 Kbit/s.

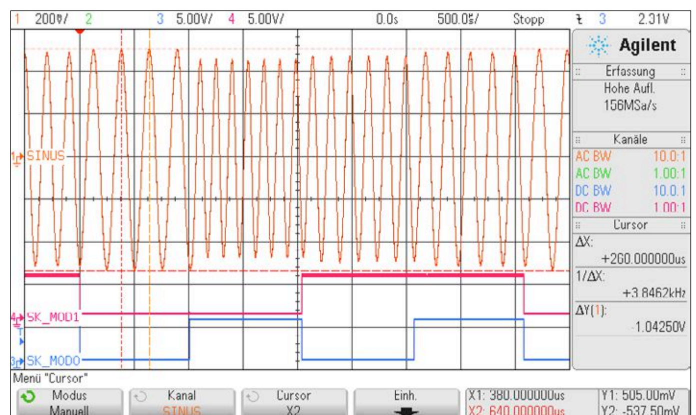
In **figuur 7** ziet u het ASK-signaal, met het bijbehorende spectrum in **figuur 8**. Hier wordt de ingang SK_MOD0 door SIM0 (afkomstig van PD2) aangestuurd. Het stimulatiesignaal SIM0 maken we intern, het ligt bij 500 Hz, wat overeenkomt met een bitrate van 1 Kbit/s. Net als bij alle andere metingen gebeurt dit onder ideale omstandigheden: het modulerende signaal is synchroon met de fase van de draaggolf.
Het spectrum is in zoverre interessant, dat het precies een helft van het spectrum van 2-FSK-modulatie voorstelt. Het bevat de frequentie van de draaggolf f_0, en de zijbanden met $f_0 \pm f_{mod}$, $f_0 \pm 3*f_{mod}$, en zo verder. We zien dus meteen dat de breedte van de zijbanden afhangt van de hoogte van de modulerende frequentie en daarmee van de data-rate. De frequentiecursors zijn zodanig gepositioneerd dat ze ongeveer 90% van het spectrum van het niet in bandbreedte begrensde signaal omsluiten. De resulterende frequentie-afstand is dus de benodigde bandbreedte van het overdrachtskanaal. OOK-modulatie wordt bijvoorbeeld toegepast in de AISG-standaard, die de aansturing van antennes voor mobiele telefonie specificeert.

DDS_FSK: Frequency Shift Keying

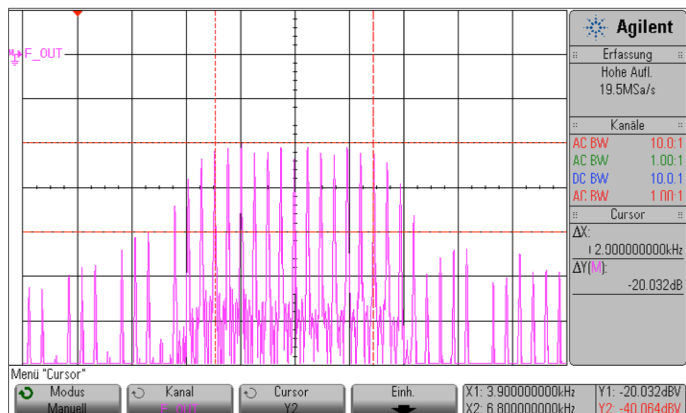
Bij FSK-modulatie wordt aan elke logische toestand van de digitale ingangsinformatie een discrete frequentie toegewezen. Met een 1-bit-ingangssignaal is er dan een frequentie voor LOW en een frequentie voor HIGH. Met een n-bit-ingang zijn er 2n frequenties nodig die aan 2n mogelijke ingangstoestanden zijn toegewezen. Elke permutatie van de n bits vormt een zogenaamd symbool. Per stap wordt er een n-bit-symbool overgedragen. Als de hoogste en de laagste frequentie constant blijven, dan is de benodigde bandbreedte van het overdrachtskanaal onafhankelijk van het aantal symbolen.
In de software is er voor elke ingangsinformatie van een 2-bits symbool een voorgedefinieerde waarde voor de variabele fase-increment gedeclareerd. De vier logische toestanden van het ingangssignaal hebben dus elk een eigen frequentie toegewezen gekregen. Met een druk op f_down op f_up kunnen deze frequenties afzonderlijk woorden geactiveerd.
Voor een eenvoudig experiment maken we een 2-bits stimulus-signaal, SIM0 en SIM1, dat als ingangssignaal naar SK_MOD_IN gaat. Doordat de toetsen via een weerstand ontkoppeld zijn, kunnen we van het 4-FSK-signaal met een druk op f_up of f_down een 2-FSK-signaal maken.
Figuur 9 geeft een 2-FSK-signaal weer, met het bijbehorende spectrum in **figuur 10**. **Figuur 11** toont een 4-FSK-signaal, met het spectrum in **figuur 12**. De spectraallijn precies in het midden van de band ontstaat doordat zijbanden elkaar overlappen. Deze overlappingen, die storing zouden kunnen geven, kunnen worden gereduceerd door de afzonderlijke frequenties handig te kiezen ten opzichte van de modulatiefrequentie. In de tijddomein-grafiek is mooi te zien dat er geen discontinuiteiten in het signaalverloop ontstaan als we van de ene naar de andere frequentie overschakelen, wat bijvoorbeeld wel het



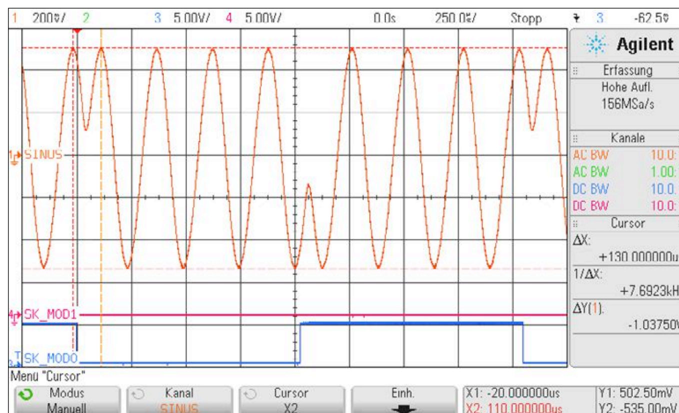
Figuur 10. Het frequentiespectrum van het 2-FSK-uitgangssignaal van figuur 9. f_{mid} = 5,37 kHz, 1 kHz/div, 10 dB/div.



Figuur 11. Het uitgangssignaal met 4-FSK.
f_1 = 3906 Hz, f_2 = 6836 Hz, f_3 = 5859 Hz, f_4 = 4883 Hz, bitrate= 2 Kbit/s.



Figuur 12. Het frequentiespectrum van het 4-FSK-uitgangssignaal van figuur 11. f_{mid}= 5,37 kHz, 1 kHz/div, 10 dB/div.



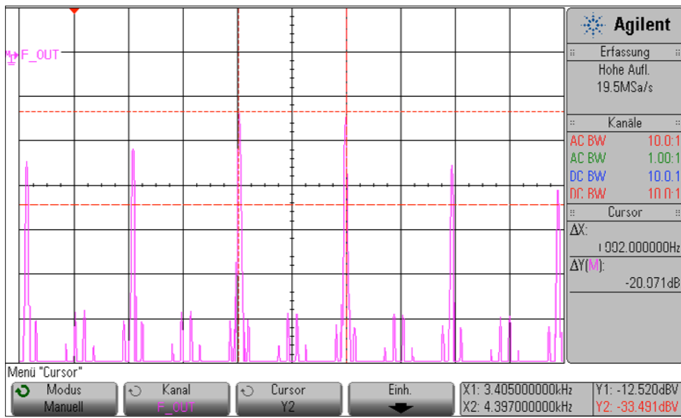
Figuur 13. Het uitgangssignaal met 2-PSK. Cursorafstand = 130 μs ≙ 180° fasesprong, bitrate= 1 Kbit/s.

geval zou zijn als we zouden overschakelen van de ene naar de andere oscillator. Dat is een groot voordeel van DDS. FSK-modulatie wordt onder andere bij faxmodems en bij DECT-telefoons toegepast.

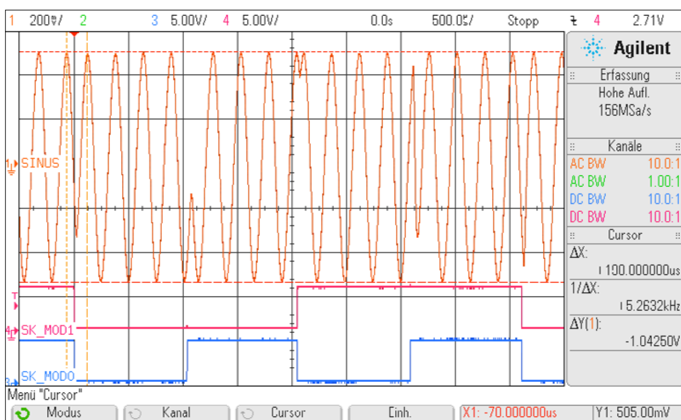
DDS_PSK: Phase Shift Keying

Bij PSK-modulatie krijgt het uitgangssignaal voor elke logische toestand van de digitale ingangsinformatie een fase-offset ten opzichte van de draaggolf. Bij elke toestandsverandering aan de ingang verandert dus de fase van het uitgangssignaal. Bij een 1-bits symbool is de fase-offset 0° voor LOW en +180° voor HIGH. Een fase-offset van −45° geeft hetzelfde resultaat als een van +135°. Om die reden gebruikt DDS_PSK die waarden: we kunnen er zowel 2-PSK als 4-PSK mee maken. Bij 2-bits symbolen is de fase-offset al naargelang het symbool 45°, 135°, 225° of 315° (≙ −45°).

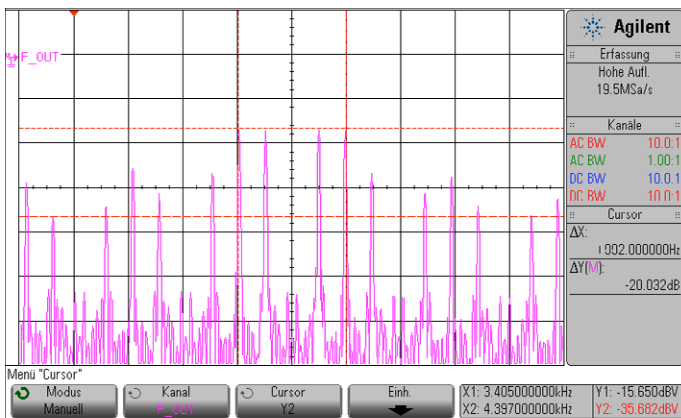
Anders dan bij FSK blijft de waarde van het fase-increment constant, omdat de frequentie stabiel moet blijven. De offsetwaarde, die afhangt van de toestand van de informatie aan de ingang, wordt opgeteld bij de fase-accumulator, wat dan weer leidt tot een pointer naar de LUT. De waarde van de fase-accumulator bevat net als eerst de momentane waarde van de



Figuur 14. Het frequentiespectrum van het 2-PSK-uitgangssignaal van figuur 13. f_{mid} = 3900 Hz, 500 Hz/div, 20 dB/div.



Figuur 15. Het uitgangssignaal met 4-PSK. Cursorafstand = 190 µs ≙ +90°; fasesprongen van +90°, 180°, -90° en 180°. Bitrate = 2 Kbit/s.



Figuur 16. Het frequentiespectrum van het 4-PSK-uitgangssignaal van figuur 15. f_{mid} = 3900 Hz, 500 Hz/div, 20 dB/div.

nul-fase van het draaggolfsignaal. De offsetwaardes voor de individuele symbolen worden gedefinieerd in een array.
Om dit te testen, leiden we weer SIM0 en SIM1 naar SK_MOD. **Figuur 13** geeft het uitgangssignaal bij 2-PSK met fasesprongen van 180°, **figuur 14** is het bijbehorende frequentiespectrum. We zien hier de draaggolf-frequentie niet; dat komt doordat de intervallen voor 0 en 1 even lang zijn binnen de tijd waarin we de meting doen. De zijbanden liggen bij $f_0 \pm f_{mod}$, $f_0 \pm 3*f_{mod}$, enzovoort.
In **figuur 15** ziet u een 4-PSK-signaal, ook wel QPSK genaamd, met sprongen van 90° en 180°. De resulterende vorm van de fasesprongen hangt af van het tijdstip waarop het symbool verandert, ten opzichte van de fasehoek van de draaggolf. **Figuur 16** toont het bijbehorende spectrum. Ook hier ziet u geen draaggolf. Dat komt, net als bij 2-PSK, door het verloop in de tijd van het modulerende signaal en het daaruit volgende verloop in de tijd van de fasehoek. De amplitude voor elk symbool is voor te stellen als een vector met de faseverschuiving uitgedrukt als een hoek. Als binnen het meetinterval de som van de met de tijdsduur vermenigvuldigde vectoren gelijk is aan nul, wordt de draaggolf onderdrukt.
De analyse van de uitgangssignalen van zowel 2-PSK- als 4-PSK-modulatie laat zien dat het terugwinnen van het digitale signaal, demodulatie dus, zonder reconstructie van de draaggolf niet mogelijk is. Bij differentiële PSK, waarbij de digitale informatie tot toestandsveranderingen is verwerkt, is dat niet het geval.
In de praktijk wordt PSK-modulatie gerealiseerd met I/Q-modulatoren, waarbij men de bandbreedte van het modulerende signaal eenvoudig kan begrenzen om de totaal benodigde bandbreedte klein te houden.
PSK-modulatie, en dan speciaal in combinatie met amplitudemodulatie, wordt ook wel QAM genoemd. Het wordt veel toegepast in mobiele communicatie, en bij digitale radio en televisie.

Conclusie

DDS is een bijzonder krachtige technologie die voor veel toepassingen inzetbaar is. Implementatie met een Arduino levert beslist geen resultaten van de allerhoogste kwaliteit, maar met weinig moeite en kosten kan men toch heel veel uitproberen en aan de weet komen. Wie zich verder in deze technologie wil verdiepen, raden we aan om de DDS Tutorial van Analog Devices [3] te bestuderen. ◄

180336-03

Weblinks

[1] “Direkte Digitale Synthese” (Duitstalig): www.elektormagazine.de/150315
[2] Projectpagina bij dit artikel: www.elektormagazine.nl/180336-03
[3] DDS-Tutorial (Engelstalig): www.analog.com/media/en/training-seminars/design-handbooks/Technical-Tutorial-DDS/technical-tutorial-DDS.pdf

elektor

NOVEMBER

Met gepaste trots presenteert Elektor International Media de Wall of Fame 2018. We werken samen met de beste bedrijven op het gebied van elektronica — van wereldwijd opererende distributeurs tot fabrikanten van onderdelen — om opwindende producten en diensten onder uw aandacht te brengen die de motor en inspiratie achter high-tech innovaties zijn. De bedrijven die dit jaar een plaatsje op de Wall of Fame hebben gekregen, bieden elektronica-engineers, makers en studenten een breed palet aan onmisbare tools en oplossingen voor het ontwikkelen van de volgende generatie elektronicaprojecten en -producten.

AVNET SILICA
www.avnet.com

EBV Elektronik | An Avnet Company |
www.ebv.com

ARROW
www.arrow.com

Honeywell
THE POWER OF CONNECTED
www.honeywell.com

KCS Research & Development Elektronica assemblage
www.kcs.tv

TRINAMIC
MOTION CONTROL
www.trinamic.com



www.icape-group.com

KEYSIGHT TECHNOLOGIES
www.keysight.com

BECKER MÜLLER
www.becker-mueller.de
www.becker-mueller.de



www.eurocircuits.com

kurtz ersa
www.kurtzersa.com

Bürklin
DIE GANZE ELEKTRONIK
www.buerklin.com

HAMMOND MANUFACTURING®
www.hammondmfg.com

CONRAD
www.conrad.com

CDPStudio
www.cdpstudio.com

HEILIND
Performance. Trust. Innovation.
www.heilind.com

LOETRONIC
www.loetronic.com

WALL OF FAME

2018



www.microchip.com



www.picotech.com



www.schaffner.com



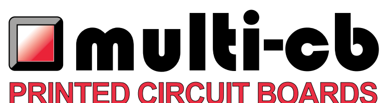
www.mouser.com



www.pi-top.com



www.schurter.com



www.multi-circuit-boards.eu



www.radiocontrolli.com



www.siglent.com



www.ni.com



www.recom-power.com



www.labcenter.com



www.peak-system.com



www.reichelt.com



www.wiznet.io



www.peaktech.de



www.lpkf.com



www.we-online.com



www.phoenixcontact.com



www.rs-online.com



www.unitronic.de

(advertorial)

Minder stroom verbruiken

door rekenen en systematisch werken

Robert Lacoste (Frankrijk)

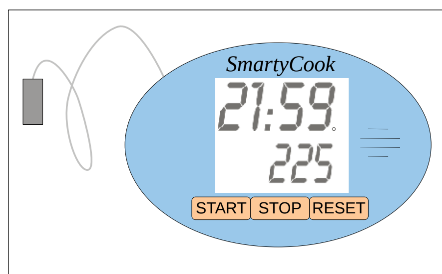


illustratie: shutterstock.com

Waarschijnlijk hebt u dezelfde ervaring als ik: dag in dag uit erger ik me aan elektronische apparaten met een accu die al na een paar dagen (of uren) leeg is of aan batterijen die al te vaak zijn overleden. En hetzelfde gebeurt misschien wel bij apparaten die u zelf hebt ontworpen. Wat kunnen we doen om de gebruiksduur van de batterijen te verlengen? Wat zijn de *best practices*? Welke tips zijn van toepassing op bijna alle projecten? Dit artikel geeft een antwoord op die vragen.

Een voorbeeld...

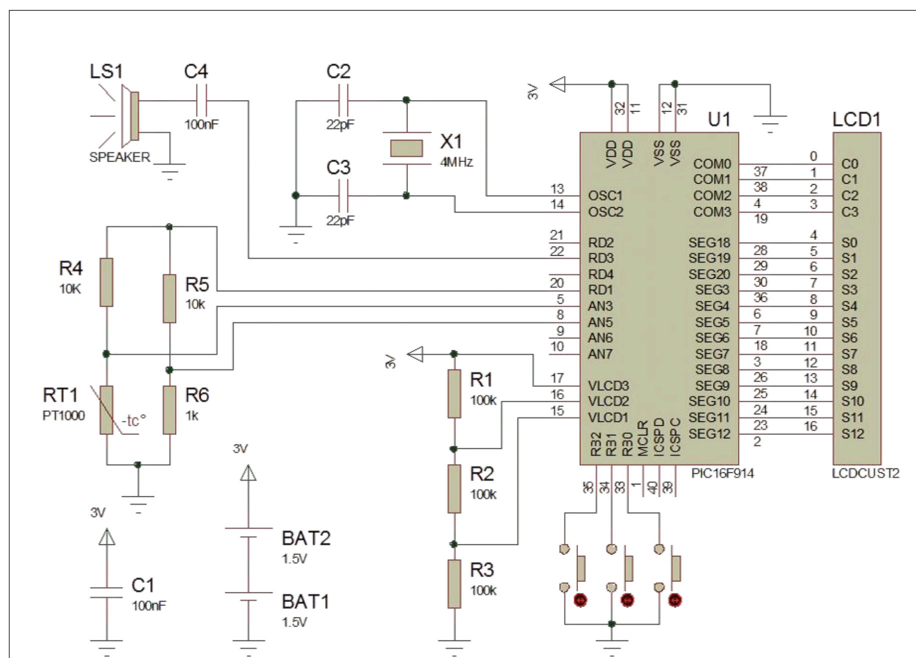
Liever dan in vage, algemene termen te praten, ga ik hier uit van een klein, concreet voorbeeld. Stel dat u een klein apparaat voor in de keuken moet ontwerpen. Het moet een timer en een thermometer hebben. Ik heb het de SmartyCook gedoopt (**figuur 1**). Er is niets ingewikkelds aan: we hebben een temperatuursensor die in het te bereiden voedsel kan worden geprikt, een paar knoppen voor het programmeren van de kooktijd of de maximale temperatuur en er is een LC-display dat het geheel tot leven brengt. Met de juiste embedded software zal uw *roastbeef* of ganzenlever smaken als in een driesterrenrestaurant. En u, als trouwe Elektor-lezer, bent heel goed in staat om zoiets in elkaar te knutselen, toch?



Figuur 1. Het voorbeeld voor dit artikel is een kleine kookwekker met thermometer.

Laten we ervan uitgaan dat u niet met een leeg vel papier begint — u hebt ergens een schema gevonden (**figuur 2**). Het spreekt vanzelf dat het project is gebaseerd op een microcontroller. Hier is gekozen voor een kleine, ultra low-power 8-bit µC: de PIC16F914 van Microchip. Deze variant van de PIC16F heeft een interface voor het gemultiplext aansturen van een LCD, waarvoor niet meer externe hardware nodig is dan twee referentiespanningen, respectievelijk gelijk aan 1/3 en 2/3 van de voedingsspanning. Deze worden geproduceerd door R1/R2/R3. Een kristal van 4 MHz zorgt voor het kloksignaal. De temperatuursensor is een thermistor van het type PT1000 (dat betekent dat hij een weerstand heeft van 1 kΩ bij 20 °C). Hij wordt gevoed door een I/O-lijn van de µC via R4. De resulterende spanning wordt gelezen

door de interne A/D-converter van de µC. Met een ander paar weerstanden wordt een referentiespanning gemaakt, zodat een brugmeting mogelijk is. Daarmee wordt voorkomen dat de voedingsspanning het meetresultaat beïnvloedt. Een kleine luidspreker zorgt voor een 'piep-piep' aan het eind van de kooktijd. Deze wordt rechtstreeks aangestuurd door een andere uitgang. En tenslotte zijn er drie drukknoppen die zijn verbonden met drie andere pinnen van de µC met interne pullup-weerstanden. De software configureert deze pinnen zodanig dat er een interrupt wordt gegenereerd als er op één van de knoppen wordt gedrukt. Het geheel wordt gevoed uit twee AAA-batterijen van 1,5 V in serie, dus met 3 V. Dat ziet er goed uit, toch? Maar let op: ik heb dit schema alleen bedacht als illustratie bij dit artikel, de schakeling is nooit in werkelijkheid opgebouwd, dus er kunnen best foutjes in zitten...



Figuur 2. Het schema van de SmartyCook vóór de optimalisatie. Helemaal niet zo ingewikkeld.

Energiebalans

Hoe moeten we nu de levensduur van de batterijen in zo'n schakeling schatten? Deze berekening, die we de energiebalans noemen, is onmisbaar voordat we de schakeling echt gaan opbouwen. Ten eerste om te kijken of die levensduur redelijk is en ten tweede om de punten te ontdekken die vatbaar zijn voor verbetering. Het opstellen van een energiebalans gaat van oudsher in vier stappen. Om te beginnen moeten we een lijst maken van de verschillende toestanden van het product. Het gaat om toestanden waarin het energieverbruik min of meer constant is. Bij de SmartyCook kunnen we drie modi onderscheiden:

- slaaptoestand (niet in gebruik);
- actief (bezig met aftellen van de tijd en meten van de temperatuur);
- produceren van het alarmsignaal (het netto stroomverbruik is dan het grootst omdat de luidspreker wordt gevoed).

Natuurlijk wordt dit onderscheid ingewikkelder naarmate een apparaat meer functies heeft. U moet ook schatten hoeveel tijd het apparaat in elke toestand zal doorbrengen, bijvoorbeeld per dag. In dit stadium moeten we uitgaan van aannames. Laten we aannemen dat de SmartyCook gemiddeld twee keer per dag zal worden gebruikt, elke keer voor een kooktijd van 15 minuten (gemiddeld genomen), en dat hij 30 seconden zal piepen voordat hij wordt gestopt. We maken dan een samenvatting van het gedrag in de verschillende toestanden, zoals in **tabel 1**.

Tabel 1: De verschillende bedrijfsmodi van de schakeling.

Modus	Omschrijving	Duur (s/dag)	Commentaar
SLAAP	Product in diepe slaap, LCD uit	84600	24h − actief
ACTIEF	Product in gebruik, LCD ingeschakeld, metingen en tellingen in gang	1800	2×15 min
PIEP	Genereren van alarmsignaal	60	2×30 s

De tweede etappe is het berekenen van het effectieve verbruik in elk van de toestanden. We gaan er vanuit dat dit constant is. We hebben alleen de datasheets van de componenten en een paar kleine berekeningen nodig. Het belangrijkste is natuurlijk om niets te vergeten en de orde van grootte juist in te schatten. De beste manier om dit te doen is stuk voor stuk te kijken naar elke verbinding in het schema, en ons af te vragen welke stroom er doorheen loopt in welke modus. In **tabel 2** zien we de resultaten voor de SmartyCook. De geschatte stromen in de drie toestanden zijn respectievelijk 12 µA, 975 µA en 10,9 mA. Het is opmerkelijk dat de µC slechts 320 µA gebruikt als hij actief is bij 3 V en 4 MHz: niet slecht!

Tabel 2: Verbruik in µA voor elk van de modi.

Modus	PIC16F914	LCD	R1/ R2/R3	R4/R5/ R6/RT1	LS	Totaal
SLAAP	2,300		10,000			12,300
ACTIEF	320,000	100,000	10,000	545,455		975,455
PIEP	320,000	100,000	10,000	545,455	10000,000	10975,455

De derde stap is de eenvoudigste, maar ook de belangrijkste: bereken het gemiddelde energieverbruik per dag, uitgedrukt in mAh. Vermenigvuldig daarvoor gewoon de tijd die de schakeling in elke toestand doorbrengt (in uren) met de stroom die er op dat moment loopt (uitgedrukt in mA) en tel dat bij elkaar op. De resultaten staan in **tabel 3**. Het blijkt dat met onze aannames de SmartyCook iets minder dan 1 mAh per dag gebruikt. Het is ook meteen te zien dat de meest gulzige toestand de modus 'actief' is (50% van het totaal). De alarmmodus vertegenwoordigt niet meer dan

Tabel 3: Gemiddeld energieverbruik per dag, uitgedrukt in mAh.				
Modus	**Duur (s/dag)**	**Momentane stroom (µA)**	**Energie (mAh)**	**% van totaal**
SLAAP	84600	12,300	0,289050	30,1%
ACTIF	1800	975,455	0,487727	**50,8%**
PIEP	60	10 975,455	0,182924	19,1%
		Totaal per dag (mAh)	**0,959702**	100,0%

Tabel 4: Levensduur van de batterijen.			
Type bron	AAA kool/zink-batterijen		
Referentie	Eveready 1212		
Capaciteit (mAh)	**Theoretisch**	**Project**	**Variatie** (schatting van de auteur)
Theoretische capaciteit (mAh)	464		
Minimale spanning (V)	1,0	1,2	20%
Levensduur (jaren)	0,1	2,0	20%
Variatie tussen leveranciers			20%
Piekstroom (mA)	16,0	10,0	-5%
Minimale temperatuur (°C)	20,0	20,0	0%
Geschatte capaciteit (mAh)			249
Dagelijks verbruik (mAh)			0,960
Geschatte levensduur (dagen)			**260**

19%. Ook al loopt er flink wat stroom in deze toestand, hij duurt maar heel kort. Onze simpele analyse heeft aangetoond welke zaken we als eerste moeten optimaliseren.
De laatste stap is het berekenen van de verwachte levensduur van de batterijen in dagen. Daarvoor hoeven we alleen maar de capaciteit van de batterij, gegeven in mAh, te delen door het dagelijkse energieverbruik, ook in mAh, om de levensduur in dagen te vinden. We gebruiken hier twee alkaline-AAA-batterijen van 464mAh (zoals gespecificeerd door de fabrikant). Het apparaat gebruikt 1 mAh per dag. Werkt het dan 464 dagen? Helaas niet: deze theoretische capaciteit volgens opgave van de fabrikant geldt slechts onder heel specifieke omstandigheden en is erg optimistisch. Zal de omgevingstemperatuur altijd 20 °C bedragen? Is de ontlaadstroom constant? Wat is de klemspanning van de batterij als hij uitgeputis volgens de fabrikant? En werkt uw apparaat nog bij die spanning? Het schatten van deze factoren is helaas nogal ingewikkeld, omdat de noodzakelijke gegevens zelden beschikbaar zijn en eerder met chemie dan met elektronica te maken hebben. Maar in de praktijk is het niet uitzonderlijk dat de werkelijke capaciteit 30 tot 50% geringer is dan de opgegeven theoretische capaciteit. Uitgaande van deze (betwistbare) veronderstellingen komen we voor de SmartyCook uit op een geschatte levensduur voor de batterijen van ongeveer 260 dagen (zie **tabel 4**). Meer dan acht maanden! Dat is toch niet zo slecht? Maar kan het misschien nog beter?

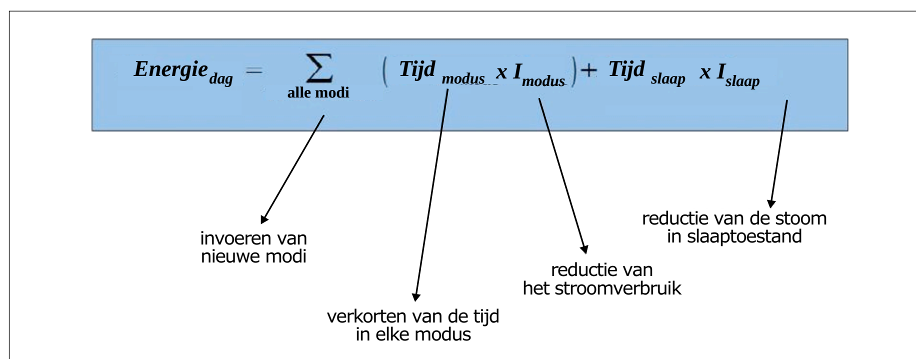
Stroomreductie

In **figuur 3** zien we de formule voor het berekenen van het dagelijkse energieverbruik. Denk eraan dat elk van de termen kan en moet worden bestudeerd voor het verlengen van de levensduur van de batterijen, en dat we daarbij systematisch te werk moeten gaan: een verbetering kan miniem lijken als een andere bijdrage duidelijk belangrijker is in de globale balans, maar kan in een later stadium essentieel blijken als die andere bijdrage is verminderd...
Laten we beginnen met de meest in het oog springende toestand: we proberen de stroom in de actieve modus te verminderen. Hoe doen we dat? De eerste oplossing, die altijd effectief is, is met alle middelen te proberen om de voedingsspanning van de elektronica te verlagen. Waarom? Vanwege de wet van Ohm, I = U/R! Als we de spanning verlagen, neemt de stroom door een weerstand evenredig af. Zelfs alle halfgeleiders gebruiken duidelijk minder stroom als de spanning wordt verminderd. Download maar eens de datasheet van de PIC16F914 [1] en werp een blik op het hoofdstuk "power consumption". Dan zult u zien dat het verbruik wordt verminderd van 320 µA tot 180 µA, een winst van 50%, als we hem voeden met 2 V in de plaats van 3 V — en dat bij gelijkblijvende prestaties! Loop dus alle componenten in uw schakeling na en gebruik altijd de laagste spanning waarbij ze nog goed samenwerken met hun omgeving. Als alle componenten kunnen werken met 2 V, zou het dom zijn om de schakeling te voeden met 3 V... Om de spanning te reduceren kan een simpele lineaire regelaar volstaan (mits we er één kiezen die zelf erg weinig energie verbruikt). Als u een kleine 2-V-regelaar van het type MPC1700 toevoegt (eigenverbruik ongeveer 2 µA) tussen de batterijen en de schakeling van de SmartyCook, neemt de gebruiksduur toe met 30%!
Sommige lezers fronsen nu misschien de wenkbrauwen: een lineaire regelaar dissipeert toch vermogen? Hoe kan die dan winst opleveren? Reken met me mee: de µC gebruikt 320 µA bij 3 V en 180 µA bij 2 V. Met een 2-V-regelaar neemt de stroom die uit de batterijen wordt getrokken af van 320 µA naar iets meer dan 180 µA (182 µA om precies te zijn), zelfs al wordt er 180 µA × (3 V – 2 V) = 180 µW gedissipeerd in de spanningsregelaar. Een zuinige DC/DC-converter zou nog meer winst opleveren, maar is wel ingewikkelder. Dat wordt het onderwerp van een volgend artikel...
Wat zijn de andere mogelijkheden? Het verminderen van de klokfrequentie is een klassieke oplossing. Bij de SmartyCook gebruiken we een kristal van 4 MHz om de µC te klokken. Denkt u werkelijk dat een µC zó snel moet draaien voor het tellen van de seconden in een kookwekker? Natuurlijk niet! Waarschijnlijk kunnen we deze frequentie een paar ordes van grootte reduceren zonder dat de gebruiker daar iets van merkt. Laten we kiezen voor een kristal van 32 kHz. Welke invloed heeft dat op het verbruik van de PIC? Dat loopt terug van 180 µA bij 4 MHz naar 8 µA bij 32 kHz! We zien dat bij alle CMOS-schakelingen: het verbruik is vrijwel

recht evenredig met de klokfrequentie. Als u alles doorrekent, zult u ontdekken dat deze drastische maatregel de gebruiksduur met slechts 10% vergroot, gewoon omdat de grootste bijdragen afkomstig zijn van andere bronnen. Maar toch is deze modificatie de moeite waard!
Tenslotte moeten we alle componenten de revue laten passeren om te kijken of er varianten van bestaan die zuiniger zijn met energie. Ook nu moeten we systematisch te werk gaan. Bestaan er modernere varianten die minder gulzig zijn? Misschien kunnen we gebruik maken van een andere technologie? Kunnen we de waarde van de weerstanden vergroten? Denkt u dat bij het voorbeeld van de SmartyCook dit soort verbeteringen mogelijk zijn? Er zijn er minstens drie. Kijk nog eens naar het schema (figuur 2) en probeer ze te vinden.
Ziet u ze? Allereerst: de temperatuursensor is een PT1000 (1 kΩ bij 20 °C). Waarom vervangen we die niet door een PT10000? Dat vermindert de meetstroom met een factor 10. Ook de drie weerstanden R1/R2/R3 voor de drie LCD-spanningen: niets verhindert ons om die te vergroten van 100 kΩ naar 470 kΩ, want de ingangsimpedanties van de μC zijn erg hoog. En hup, alweer een paar μA gewonnen. Tenslotte: is er geen betere technologische oplossing dan een luidspreker? Een piëzo-transducer is heel geschikt: hij maakt evenveel geluid en verbruikt duidelijk minder stroom.
Denkt u dat dat maar kleine verbeteringen zijn? Hier geldt, zoals zo vaak: vele kleintjes maken één grote. Als we alle kleine verbeteringen in aanmerking nemen en opnieuw de energiebalans berekenen, vinden we dat de levensduur is toegenomen van 260 dagen tot... 1055 dagen! Niet gek toch?

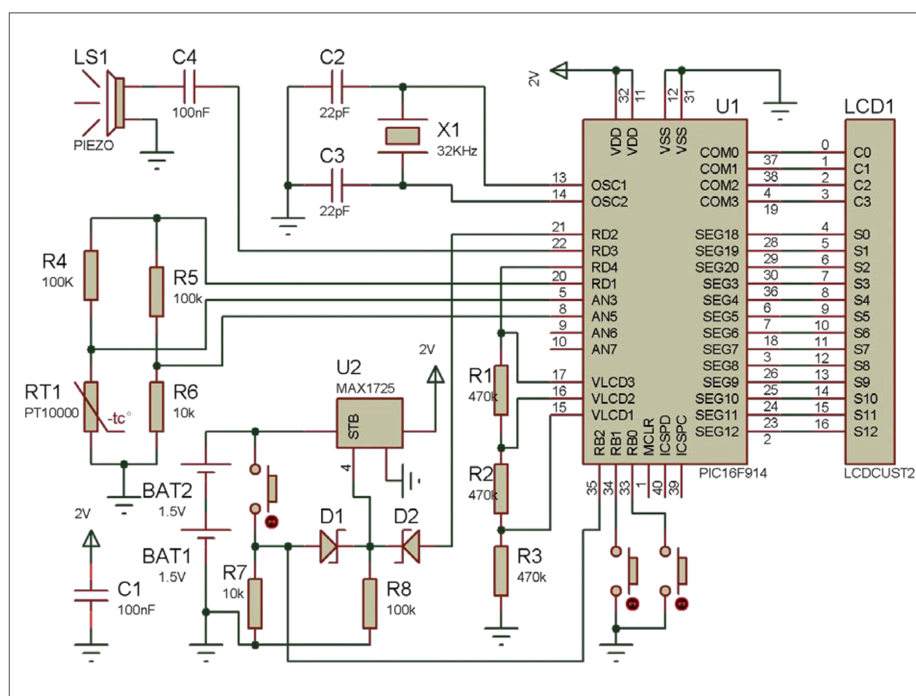
Verminderen van de inschakelduur

Maar we zijn er nog niet. We weten dat de verbruikte energie evenredig is met de stroom, maar ook met de tijd gedurende welke die stroom loopt. We kunnen dus ook veel winnen door de tijd die de schakeling doorbrengt in de meest gulzige toestanden te verkorten, of door modi op te splitsen in verschillende delen die gemiddeld minder stroom gebruiken. Enkele klassieke voorbeelden? Een oplossing die ik heel mooi vind is het apparaat efficiënter te maken: als we hetzelfde resultaat kunnen bereiken in minder tijd, gaat het apparaat sneller terug naar de



Figuur 3. Elke term die bijdraagt aan het energieverbruik kan een mogelijkheid voor optimalisatie bieden.

rusttoestand. Dat is winst voor de batterij, maar ook voor de gebruiker.
En over traditionele methodes gesproken: ook het gebruik van meer interrupts, zodat de μC vaker in de slaaptoestand kan blijven, maakt verschil. Als we een vertraging willen realiseren, is het natuurlijk beter om gebruik te maken van een *timer* en de μC in slaaptoestand te zetten tot de timer afloopt, dan om hem doelloos rond te laten lopen in een wachtlus terwijl de batterij wordt leeggetrokken. Gegevens comprimeren voordat we ze verzenden is ook een effectieve manier voor het reduceren van de tijd die het verzenden kost. Bijvoorbeeld het optimaliseren van een communicatieprotocol door data binair te verzenden in plaats van in ASCII, kan veel energie besparen.
Denk ook aan de kwaliteit van de code zelf: bij één van mijn projecten heb ik het energieverbruik met bijna 20% verminderd door gewoon een betere compiler te gebruiken voor het produceren van de code!
Wat zouden we kunnen doen in het voorbeeld van de SmartyCook? Misschien korte piepjes te gebruiken in plaats van een langgerekte pieptoon? Als we de transducer elk 2 s gedurende 0,1 s activeren, blijft het signaal hoorbaar maar de actieve ‘piep’-toestand wordt verkort met een factor 20. De berekening laat zien dat dit een enorme impact heeft op de levensduur van de batterijen, want deze toestand was, na alle andere verbeteringen, inmiddels één van de grootste verbruikers geworden: met alleen deze verbetering loopt de levensduur op van 1000 dagen naar 1500 dagen!



Figuur 4. Schema van de SmartyCook na de optimalisatie. Er lijkt niet veel verschil met het initiële schema, maar de levensduur van de batterijen is vijftien keer zo groot geworden!

Een andere *best practice* is om ons bij alle componenten af te vragen of het werkelijk noodzakelijk is om ze in alle toestanden te voeden. Kijk opnieuw naar het schema van de SmartyCook, in het bijzonder naar R1/R2/R3. Ziet u ruimte voor een grote verbetering? Die drie weerstanden zijn rechtstreeks verbonden met de batterijen, en gebruiken dus continu stroom. Maar ze zijn alleen nuttig als het LCD actief is. We kunnen dus hun voeding uitschakelen als het product in rust is, gewoon door ze niet rechtstreeks te voeden, maar via een uitgang van de µC. Een kleine aanpassing, maar hij heeft wel een grote impact, want de verwachte levensduur gaat nu van 1500 dagen naar 2391 dagen...

Tabel 5: Verbruik in µA in elk van de bedrijfsmodi na optimalisatie.

Modus	MAX-1725	PIC-16F914	LCD	R1/R2/R3	R4/R5/R6/RT1	LS	Totaal
SLAAP	0,700						0,700
ACTIEF	2,000	6,000	80,000	1,418			89,418
METEN	2,000	6,000	80,000	1,418	36,364		125,782
PIEP	2,000	8,000	80,000	1,418		250,000	341,418

Tabel 6: Gemiddeld energieverbruik per dag, uitgedrukt in mAh.

Mode	Duur (s/dag)	Momentane stroom (µA)	Energie (mAh)	% van totaal
SLAAP	84600	0,700	0,016450	24,6%
ACTIEF	1800	89,418	0,044709	**66,8%**
METEN	2	125,782	0,000063	0,1%
PIEP	60	341,418	0,005690	8,5%
Totaal per dag (mAh)			**0,066912**	100,0%

Tabel 7: Levensduur na optimalisatie.

Capaciteit (mAh)	Theoretisch	Project	Variatie (schatting van de auteur)
Theoretische capaciteit (mAh)	464		
Minimale spanning (V)	1,0	1,2	20%
Levensduur (jaren)	0,1	2,0	20%
Variatie tussen leveranciers			20%
Piekstroom (mA)	16,0	10,0	-5%
Minimale temperatuur (°C)	20,0	20,0	0%
Geschatte capaciteit (mAh)			249
Dagelijks verbruik (mAh)			0,067
Geschatte levensduur (d)			**3 728**

Reduceren van de standby-stroom

Er is nog een laatste manier om te optimaliseren: sparen op de µA's die worden verbruikt als het product in rust is. Schakel alles uit wat uitgeschakeld kan worden. Daar kunnen we heel ver in gaan. Bijvoorbeeld bij onze SmartyCook: de µC wordt de hele tijd gevoed. Waarom? Omdat we hem met één druk op de knop willen wekken. Er bestaat een klassiek trucje om de voeding van de schakeling in rust helemaal uit te schakelen. Kijk maar eens naar **figuur 4**, het geoptimaliseerde schema van de SmartyCook. Ik heb een lineaire regelaar met een 'shutdown'-ingang (U2, MAX1725) toegevoegd. Een druk op de knop activeert deze ingang, en dus de voeding van de µC. En dan schakelt de software uitgang RD2 in, die de regelaar ingeschakeld houdt via de OR-functie die is gerealiseerd met D1 en D2. De schakeling werkt dus normaal zolang de µC de *shutdown*-pen niet weer via RD2 laag maakt. Doet hij dat wel, dan schakelt de voeding uit: 'zelfmoord'. De schakeling gebruikt nu vrijwel geen stroom meer tot de volgende druk op de knop.

Conclusie

Als u het oorspronkelijke schema vergelijkt met dat van figuur 4, ziet u dat de aanpassingen maar klein zijn: hier en daar is een weerstandswaarde veranderd, enkele punten zijn geoptimaliseerd en de voeding is iets ingewikkelder geworden. Maar de theoretisch verwachte levensduur van de batterijen is radicaal veranderd. Vergelijk de nieuwe energiebalans (**tabellen 5...7**) maar eens met de eerste: de gebruiksduur is van 260 dagen naar 3728 dagen gegaan, dat is van acht maanden naar meer dan tien jaar (als de zelfontlading van de batterijen tenminste niet te groot is...)

Natuurlijk is dit voorbeeld speciaal bedacht om heel duidelijk te demonstreren hoe de methode werkt, maar ik hoop dat ik u ervan overtuigd heb dat het vaak mogelijk is om de energiebalans te verbeteren. Ik heb ook geprobeerd u te laten zien dat, ook al zijn de adviezen eenvoudig, we de optimalisatie wel methodisch en systematisch moeten aanpakken. Dankzij zo'n manier van werken bereiken veel producten op de markt een opmerkelijke batterij-levensduur. Ik heb onder andere gewerkt aan allerlei projecten voor gas- en watermeters: die apparaten worden gevoed met batterijen en kunnen werkelijk 15 of 20 jaar functioneren terwijl ze meerdere keren per dag draadloos berichten uitzenden over een afstand van enkele kilometers... Het is dus echt mogelijk.

En nu bent u aan zet! Ik raad u aan om uw laatste project nog eens onder de loep te nemen. En probeer dan de methode uit dit artikel eens toe te passen. Slaagt u erin om een grote verbetering te realiseren? In het volgende artikel zal ik in detail ingaan op een andere techniek die erg nuttig is voor energetische optimalisatie: schakelende spanningsomzetters.

(160633)

Dit artikel is eerder verschenen in Circuit Cellar 213 (april 2008).

Link

[1] Datasheet van de PIC16F914 van Microchip: www.microchip.com/wwwproducts/en/PIC16F914

AVO buizentesters

" still going strong "

Dr. ir. Martin Beusekamp

De Historische Studieverzameling van de faculteit Elektrotechniek, Wiskunde & Informatica (EWI) van de Universiteit Twente [1] omvat meer dan 1100 apparaten uit de geschiedenis van de vakgebieden van de faculteit: elektronische meetinstrumenten, computers, rekenmachines, telecommunicatieapparatuur, etc. Deze apparatuur wordt beheerd door vrijwilligers, overwegend gepensioneerden van de faculteit. Deze aflevering van Retrotronica gaat over buizentesters van de firma AVO, waarvan er enkele deel uitmaken van de betreffende studieverzameling.



Een AVO buizentester in volle glorie. Let op het bijgeleverde handboek in de houder onder het apparaat. Fotografie: Gerrit Busscher.

Een buizentester is niets meer en niets minder dan een aantal voedingen, een flinke reeks schakelaars en potentiometers, een paneel met verschillende buisvoeten en een meetinstrument, samengebracht in één behuizing (zie de foto hierboven). In zijn algemeenheid moet een buizentester vier verschillende spanningen kunnen aanbieden aan de te testen buis: de gloeispanning (zie ook het kader **'Alleen voor gloeispanningsbuizen?'**), de anodespanning, de schermroosterspanning en de (negatieve) voorspanning op het stuurrooster. Niet altijd zijn alle spanningen nodig: een diode heeft geen roosters, een triode geen schermrooster, een buis met koude kathode geen gloeidraad, etc.
In een AVO buizentester worden die vier spanningen verzorgd door drie gescheiden voedingen, zelfs alle drie met een eigen netspanningstransformator. Laten we die voedingen, de kern van alle buizentesters, eens aan een nadere beschouwing onderwerpen.

Gloeispanning nodig?

De gloeispanningsvoeding van een AVO buizentester is een wisselspanningsbron die veel stroom moet kunnen leveren. De AZ4, bijvoorbeeld, een dubbeldiode in de voeding van veel buizenradio's uit het midden van de vorige eeuw, trekt 2,3 A gloeistroom bij een gloeispanning van 4 V. Sommige industriele buizen gebruiken nog wel hogere gloeistromen.
De kathode van een elektronenbuis is gemaakt van een materiaal waaruit bij verhitting relatief makkelijk vrije elektronen kunnen ontsnappen (thermische emissie, ook wel Edisoneffect genoemd). Die verwarming gebeurt door een gloeistroom door de kathode te laten lopen (directe verhitting) of door een aparte gloeidraad in de onmiddellijke nabijheid van de kathode (indirecte verhitting).
Om die gloeistroom te laten lopen, is een bepaalde gloeispanning nodig (zie weer het kader "Alleen voor gloeispanningsbuizen?"). En zo veel buizen, zo veel verschillende gloeispanningen. Vandaar dat een AVO buizentester maar liefst 121 verschillende gloeispanningen kan leveren. Een 11-standenschakelaar voor de grove selectie levert een gloeispanning van 0 – 110 VAC. Nog een 11-standenschakelaar voor de fijne selectie telt daar 0 – 7,5 VAC bij op.
De laagste gloeispanning die een AVO buizentester kan leveren is 0,625 VAC, de hoogste bedraagt 117,5 VAC, nagenoeg gelijk aan de netspanning in de Verenigde Staten in de eerste decennia na de Tweede Wereldoorlog. Inderdaad bestonden daar elektronenbuizen waarvan de gloeidraad rechtstreeks op het lichtnet kon worden aangesloten (bijvoorbeeld de types 117Z3 en 117Z6).

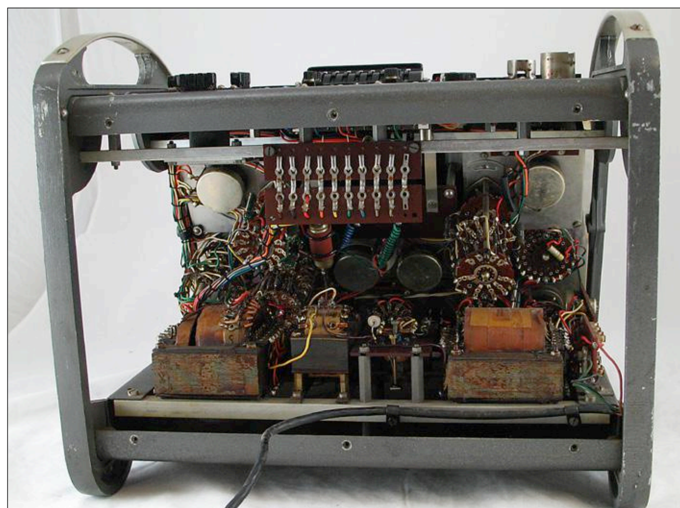
Twee flinke spanningen

Aan een wolk vrije elektronen om de kathode hebben we nog niet zo veel. Wel als we die elektronen naar de anode trekken met een relatief hoge gelijkspanning op die anode. De negatief geladen elektronen bewegen van kathode naar anode, de gelijkstroom loopt dus van anode naar kathode. Omgekeerd werkt het niet. Om de anode hangt geen wolk van vrije elektronen, in de andere richting kan dus geen stroom lopen. Zie-

daar, de elektronenbuis als diode!
Een AVO buizentester heeft een 17-standenschakelaar voor anodespanningen van 12,6 – 400 V. Een fijnregeling ontbreekt, want die is ook niet echt nodig. Wie per se de eigenschappen van een buis wil weten bij een anodespanning van 275 V (niet voorhanden op de keuzeschakelaar), doet een meting bij 250 V en een meting bij 300 V en neemt het gemiddelde daarvan.
De relatief hoge anodespanning die de vrije elektronen rond de kathode aantrekt, geeft die elektronen een flinke snelheid. Sommige kaatsen daarom terug van de anode en verstoren de werking van de buis. Vandaar dat buizen met meer dan één rooster (tetrodes, pentodes, etc.) een schermrooster tussen kathode en anode hebben op een relatief hoge spanning. Dat rooster 'vangt' ook sommige elektronen die nog onderweg zijn van kathode naar anode, maar de meeste daarvan hebben voldoende snelheid om het schermrooster te passeren. Bijna alle teruggekaatste elektronen worden wél door het schermrooster opgenomen.
Dezelfde voeding die de anodespanning verzorgt, verzorgt in AVO buizentesters ook de schermroosterspanning in 17 stappen van 12,6 – 300 V. Ook hier ontbreekt een fijnregeling, om dezelfde reden als hiervoor genoemd.

Wel spanning, geen stroom

Het belangrijkste rooster in elektronenbuizen is het stuurrooster, het eerste rooster vanaf de kathode. Het stuurrooster (grid 1; g1) staat op een negatieve spanning ten opzichte van de kathode en remt de elektronenstroom dus af. Met een voldoende hoge negatieve spanning op g1 kan de stroom door de buis zelfs helemaal tot nul worden gereduceerd.
Juist omdat de stuurroosterspanning negatief is, stoot het stuurrooster elektronen af en loopt er dus geen stuurroosterstroom, afgezien van een klein stroompje om de capaciteit van het rooster op te laden of te ontladen bij een verandering van de stuurroosterspanning. Vreemd idee, een voeding die wel een instelbare spanning moet kunnen leveren, maar geen stroom. Toch kan een AVO buizentester een stuurroosterstroom van maximaal 100 µA meten. Als er een aanzienlijke stuurroosterstroom optreedt, duidt dit op een buis die niet meer vacuüm is, de gemeten stroom wordt veroorzaakt door binnengelekt gas.



Figuur 1. Het inwendige van een AVO buizentester. Fotografie: Gerrit Busscher.

De stuurroosterspanning wordt in een AVO buizentester opgewekt met de derde voeding. Het bereik is 0 – 100 V en hier is behalve een stappenschakelaar wél voorzien in een fijnregeling met een potentiometer om alle gewenste stuurroosterspanningen te kunnen aanbieden.

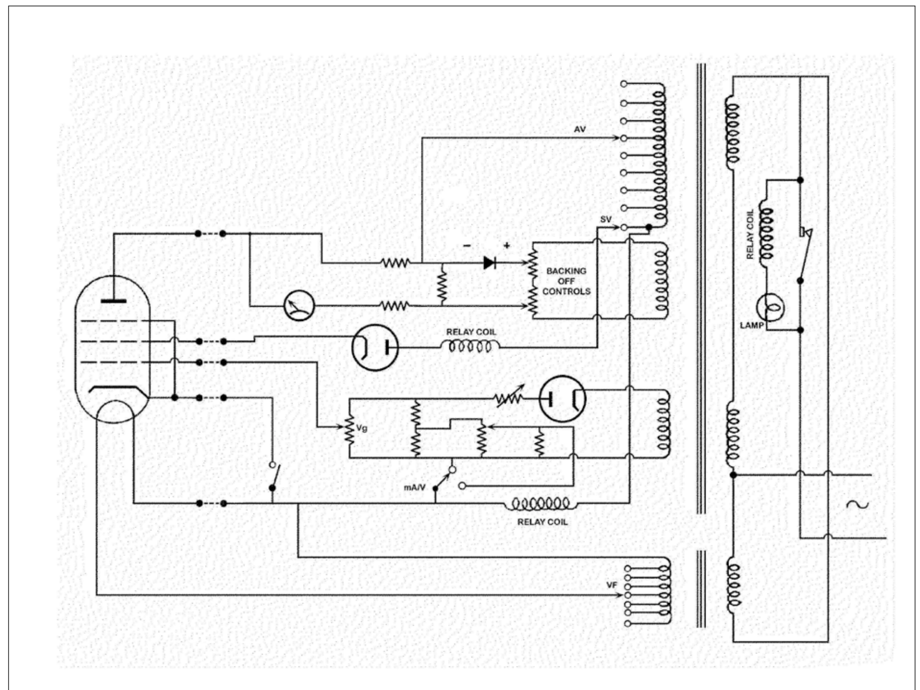
Waar zitten die elco's dan?

Figuur 1 toont het inwendige van een AVO buizentester. Duidelijk zijn de drie trafo's te zien van de drie gescheiden voedingen die we aan het begin al noemden, plus een flink aantal schakelaardekken voor de grove instelling en potentiometers voor fijnregelingen. Maar waar zitten toch de elco's die van de anodespanning, schermroosterspanning en stuurroosterspanning mooie, 'gladde' gelijkspanningen maken?
Het antwoord is simpel, die elco's zijn er niet. Om kosten te besparen zijn de anodespanning, de schermroosterspanning en de stuurroosterspanning in een klassieke AVO buizentester enkelfasig gelijkgerichte, maar niet afgevlakte of

Alleen voor gloeispanningsbuizen?

AVO buizentesters kunnen alleen een gloei**spanning** aanbieden aan de buis die getest moet worden. Alle Europese buizen waarvan het typenummer begint met een A (4 V), een D (1,4 V) of een E (6,3 V) kunnen dus probleemloos getest worden, om maar een paar voorbeelden te noemen.
Maar uiteraard bestaan er ook veel buizen die bedoeld zijn voor een gloei**stroom**. Zoals buizen waarvan het typenummer begint met een P (300 mA) of een U (100 mA). Immers, in de meeste buizentelevisies stonden de gloeidraden van alle buizen met elkaar in serie en waren ze via een serieweerstand of -condensator rechtstreeks aangesloten op het lichtnet.
Kunnen die buizen dan niet getest worden? Natuurlijk wel! In alle buizen is de gloeidraad – eenmaal op temperatuur – niets meer en niets minder dan een gewone weerstand. Een gloeispanning laat dus een gloeistroom vloeien en omgekeerd veroorzaakt een gloeistroom een spanningsval over de gloeidraad. Daarom zaten er in buizentelevisies soms toch buizen als de ECC82 (12AU7) en de EF80 (6BX6). Bedoeld voor 6,3 V gloeispanning, maar toevallig met een gloeistroom van 300 mA, dus toch geschikt voor serieschakeling met P-buizen.
Neem bijvoorbeeld een bekende vermogenspentode als de PL500 met een gloeistroom van 300 mA. In ieder buizenboek (en tegenwoordig op internet) is te vinden dat die gloeistroom een spanningsval van 27 V over de gloeidraad bewerkstelligt. Op de AVO buizentester kiezen we dus een gloeispanning van 27 V. We maken daarbij gebruik van het feit dat er in een buizentester altijd maar één buis tegelijk getest wordt en we ons dus niet hoeven af te vragen of de gloeidraden parallel of in serie geschakeld zijn.

gestabiliseerde 50-Hz wisselspanningen! In de types Mark III en eerder was de anodespanning zelfs een in het geheel niet gelijkgerichte (dus full-wave) 50-Hz wisselspanning!
AVO buizentesters maken gebruik van het feit dat de negatieve helft van de sinus van de anodespanning helemaal niets doet door de diodewerking van de buis. In het sterk vereenvoudigde schema in **figuur 2** van een buizentester type Mark III is dan ook te zien dat de anodespanning AV rechtstreeks aan de transformator wordt onttrokken en dus een wisselspanning is met een positieve en een negatieve sinushelft. Aan de richting van de dioden voor de schermroosterspanning SV en de stuurroosterspanning Vg is te zien dat de schermroosterspanning een positieve sinushelft en de stuurroosterspanning een negatieve sinushelft is. Alle spanningen die aan de buis worden aangeboden, variëren dus voortdurend, maar hebben 'gemiddeld' de gewenste ingestelde waarde. En de traagheid van het meetinstrument zorgt er voor dat daarop ook de gemiddelde waarde van de gemeten grootheid wordt aangegeven.



Figuur 2. Sterk vereenvoudigd schema van een AVO buizentester.

Als voor een bepaalde meting aan een buis niet de gemiddelde waarde van belang is, maar de effectieve waarde van de te meten grootheid, bijvoorbeeld de anodestroom, wordt daar in de ijking van het meetinstrument rekening mee gehouden. Immers, ten opzichte van de topwaarde V_t is de gemiddelde waarde van een sinushelft:

$$2 / \pi (V_t) = 0{,}636\ V_t$$

en de effectieve waarde is

$$½ \sqrt{2} (V_t) = 0{,}707\ V_t.$$

Voor de verhouding 0,707 / 0,636 = 1,110 is bij de betreffende meting eenvoudig te compenseren.
Deze meetmethode met halve en hele sinussen in plaats van gelijkspanningen heeft AVO geen windeieren gelegd. Behalve een besparing in componenten in hun eigen meetapparatuur heeft AVO het idee vastgelegd in een Brits patent [2] en er ook langs die weg nog wel iets aan verdiend.

Steilheid

Dat de anodestroom gewijzigd kan worden met de stuurroosterspanning (zie: 'Wel spanning, geen stroom') is precies de werking van de buis als versterker. De belangrijkste eigenschap van een buis is dan ook zijn 'steilheid', de verandering van de anodestroom in milliampère (mA) bij een gegeven wijziging van de stuurroosterspanning in volt (V). De steilheid wordt dus uitgedrukt in mA/V en de naam is afgeleid van de steilheid van de grafiek van anodestroom (verticaal) als functie van de stuurroosterspanning (horizontaal). Uiteraard kunnen alle AVO buizentesters de steilheid van een buis meten.
Daarbij komt de nauwkeurigheid van de meting om de hoek kijken. Stel dat een buis bij een zeker instelpunt een anodestroom van 50 mA voert en we de steilheid willen meten bij een klein signaal, dus bij een kleine variatie op de stuurroosterspanning. Als de variatie van de anodestroom bijvoorbeeld 1 mA bedraagt, hebben we het over slechts 2% variatie in de anodestroom. Daarom geeft een AVO buizentester ons de mogelijkheid om rond het DC-instelpunt van de buis een 'backing-off' stroom in te stellen (zie het schema in figuur 2) die de anodestroom rond het instelpunt compenseert. Daardoor

Weblinks

[1] Historische Studieverzameling van de faculteit Elektrotechniek, Wiskunde & Informatica (EWI) van de Universiteit Twente: http://studieverzameling.utwente.nl/

[2] Brits patent verleend aan AVO: https://frank.pocnet.net/instruments/AVO/HR/ValveTesters/patent/AVOpatent.pdf

[3] AVO Mk4 buizentester, servicehandboek: https://frank.pocnet.net/instruments/AVO/HR/ValveTesters/manuals/VCM_Mk-IVservice.pdf

[4] AVO Mk4 buizentester, gebuikershandboek: https://frank.pocnet.net/instruments/AVO/HR/ValveTesters/manuals/VCM_Mk-IVuser.pdf



Figuur 3. Geschikt voor buizen met alle gangbare voeten. Fotografie: Gerrit Busscher.



Figuur 4. Het carrousel van duimwielschakelaars, bij iedere buis hoort een unieke code. Fotografie: Gerrit Busscher.

kan de stroommeter van het apparaat naar een gevoeliger bereik geschakeld worden en kan de steilheid nauwkeuriger gemeten worden.

Buisvoeten, buisvoeten, ...

Bovenop een AVO buizentester is een groot aantal verschillende buisvoeten gemonteerd, bij het type Mark IV zijn het er bijvoorbeeld negentien (**figuur 3**). Van al die buisvoeten zijn alle contacten nummer 1 met elkaar verbonden, evenals alle contacten nummer 2, etc. Uiteraard is er een aparte draad meegeleverd met een soort krokodillenklem voor buizen met een topcontact aan de ballon.

Nu hoeven we nog slechts alle hierboven besproken spanningen aan te bieden op de juiste pen van de buis die getest moet worden. Dat gebeurt met het carrousel van duimwielschakelaars midden tussen de buisvoeten (**figuur 4**). Zie **het kader 'Handboek altijd nodig'** hoe dat precies in zijn werk gaat.

Parasitaire oscillaties

Iedere AVO buizentester bevat vele tientallen meters montagedraad om alle hiervoor besproken voedingen, schakelaars en buisvoeten op de juiste manier met elkaar te verbinden. Vanzelfsprekend hebben al die draden hun eigen parasitaire zelfinductie en parasitaire capaciteit ten opzichte van andere draden. Oscillatieneigingen van de buis die getest wordt liggen dus op de loer.

AVO buizentesters testen echter geen hoogfrequenteigenschappen. Alle bedrading kan daarom worden uitgevoerd met ferrietkralen om parasitaire oscillaties te onderdrukken. Met name de bedrading tussen de verschillende buisvoeten en rondom het carrousel van duimwielschakelaars is toegerust met ferrietkralen. Wel waarschuwt de servicemanual [3] er uitdrukkelijk voor om na eventuele reparaties alle bedrading weer terug te leggen in zijn oorspronkelijke positie.

Handboek altijd nodig!

De meeste lezers van dit blad zijn ongetwijfeld in staat om een universeelmeter of een voeding te bedienen zonder eerst de handleiding van het apparaat te bestuderen (niet altijd verstandig, overigens...). En bij een functiegenerator of een oscilloscoop zal het meestal ook nog wel lukken zonder schriftelijke ondersteuning.

En bij een AVO buizentester? Vergeet het maar! Vóór de buis getest kan worden, zal het handboek geraadpleegd moeten worden. Immers, we moeten er voor zorgen dat alle spanningen (gloeispanning, anodespanning, schermroosterspanning, negatieve stuurroosterspanning) op de juiste pennen van de buisvoet worden aangeboden. En de nummering van die pennen is bij alle buizen weer verschillend.

Dat wordt in alle AVO buizentesters geregeld door middel van een 'carrousel' van duimwielschakelaars bovenop het apparaat (figuur 4). Vóór het inschakelen van de netspanning van de buizentester moet die reeks van schakelaars op de goede cijfercode staan (voorbeeld: voor de EF80 is de code '141 230 651'). Die code staat per buistype in het handboek, dat je er dus altijd bij nodig hebt. Je moet er immers niet aan denken dat de anodespanning van (bijvoorbeeld) 250 V op de gloeidraad komt te staan. Daarom was bij de eerste series AVO buizentesters het handboek zelfs met een kettinkje aan het apparaat vastgemaakt. In de kopfoto van het artikel is het handboek in een houder onder het apparaat te herkennen.

Overigens zijn AVO buizentesters wel intern beschermd tegen schade door te hoge stromen uit de voeding voor de anode-, schermrooster- en stuurroosterspanningen. Boven een bepaalde kritische stroomwaarde spreekt een relais aan die de buizentester uitschakelt — zie de relaisspoelen in figuur 2, de anodestroom is door weerstanden begrensd. Bedieningsfouten kunnen overigens wél het einde betekenen van de buis die getest wordt, net zoals je met een curvetracer een diode of transistor kunt vernielen als je spanningen, stromen of vermogens aanbiedt waar die component niet blij van wordt.

ESTD 2004

www.elektor.tv 

Retrotronica is een rubriek over oude elektronica en legendarische Elektor-ontwerpen. Bijdragen, suggesties en vragen zijn meer dan welkom; stuur uw telex of telegram naar **redactie@elektor.nl**

Oerdegelijk

Zo degelijk als de AVO multimeters zijn, zo degelijk zijn ook de AVO buizentesters. De Historische Studieverzameling EWI heeft er drie, waarvan één in draagbare uitvoering (**figuur 5**). Van dat draagbare exemplaar is helaas het meetinstrument defect. Revisie van de beide tafelmodellen, een type Mark III en een type Mark IV, was simpel. Schoonmaken, schakelaars en potmeters smeren, nieuw netsnoer monteren en draaien maar! Het was zelfs niet nodig om bij de eerste test de netspanning met een regeltransformator langzaam (gedurende meerdere uren!) op te voeren voor het opnieuw formeren van de elektrolytische condensatoren die lang spanningsloos geweest waren. De apparaten bevatten immers geen elco's!

Voor alle gebruikers

In het buizentijdperk was het heel gewoon om van – met name professionele – apparatuur tijdens een onderhoudsbeurt 'zwakke' buizen te vervangen. Voor snelle servicebeurten heeft de paneelmeter van een AVO buizentester een rode en een groene sectie voor een simpele goed/foutindicatie van de betreffende buis (zie de kopfoto). Voor ontwerpers en ontwikkelaars die veel dieper in de karakteristieken van een buis willen duiken, biedt de gebruikershandleiding [4] een keur aan verschillende metingen die met de tester kunnen worden



Figuur 5. Ook in draagbare versie beschikbaar. Fotografie: Gerrit Busscher

AVO?
Waar kennen we die ook alweer van?

De Britse firma AVO Ltd., oorspronkelijk opgericht in 1923 onder een andere naam, is ongetwijfeld het meest bekend vanwege hun oerdegelijke universeelmeters met leren draagriem en een oorspronkelijk eikenhouten, later bakelieten behuizing (**figuur 6**). De naam AVOmeter™ betekent niet voor niets 'Ampères, Volts & Ohms (meter)'. Dat was precies wat ze konden… meten!
Oorspronkelijk gevestigd aan Vauxhall Bridge Road in het hart van Londen, ongeveer een kilometer van Buckingham Palace, heeft AVO de bekende universeelmeters met twee draaiknoppen en een 'niervormig' scherm, maar liefst 85 jaar lang geproduceerd (1923 – 2008). Tegenwoordig maakt AVO deel uit van de Megger Group Ltd. Erg bekende producten van AVO zijn echter ook hun testers voor elektronenbuizen.



Figuur 6. Oerdegelijk, die AVO multimeters (voorbeeld uit ca. 1960).

uitgevoerd, inclusief ongewenste lekstromen tussen alle elektroden, zowel bij koude als bij warme kathode.
Geen wonder dat deze klassieke AVO buizentesters decennialang populaire meetinstrumenten zijn geweest. En nog steeds zijn, getuige de prijzen van ruim duizend euro voor gebruikte exemplaren op eBay. ◂

(180463-01)



ELEKTOR ETHIEK

Kunstmatige intelligentie in het wild

Tessel Renzenbrink (Nederland)

Kunstmatige Intelligentie (Artificial Intelligence) doet de fantasie op hol slaan. De term heeft een hoog science fiction-gehalte. Als het ter sprake komt, gaat het al gauw over Terminators en robot-overlords. Maar dat soort speculaties leidt af van de werkelijke stand van zaken. Hoe ziet de toepassing van Kunstmatige Intelligentie (KI) er in de praktijk uit? De AI Expo die 27 en 28 juni plaatsvond in Amsterdam, bood een kijkje achter de schermen. Deze handelsbeurs richt zich op bedrijven die geïnteresseerd zijn in 'praktische implementatie van Artificial Intelligence en Big Data' in de bedrijfsvoering [1]. Sprekers, exposanten en bezoekers deelden verhalen over het gebruik van KI op de werkvloer. Wat vooral opviel is dat de Big Data waarmee het KI-beest gevoed moet worden, ontzettend moeilijk te verzamelen is. Afgaande op die verhalen had de beurs misschien beter de uitdaging-van-het-stroomlijnen-van-data expo kunnen heten.

Julio Peironcely illustreerde deze problematiek aan de hand van een praktijkvoorbeeld (de digitalisering van luchthaven Schiphol). Peironcely is hoofd van het Data Science-team van Schiphol. Het doel van zijn afdeling is om met data de luchthaven veiliger, efficiënter en gebruiksvriendelijker te maken. Eén project bestaat uit het optimaliseren van de turnaround-tijd ('omdraaitijd'): de tijd die het kost om een vliegtuig na aankomst weer klaar te maken voor vertrek. Tijdens dit proces worden meerdere activiteiten uitgevoerd, zoals het uitladen van vracht, het ontschepen van passagiers, bijtanken en catering aanvullen.

De man met de verrekijker

Op dit moment is er geen real-time data beschikbaar over het omdraaiproces. Daardoor weet de verkeerstoren niet precies wanneer de gate weer beschikbaar zal zijn. Peironcely laat een plaatje zien van een man in een verkeerstoren die met een verrekijker visueel checkt of een vliegtuig de gate bezet houdt. Dat plaatje is van een Amerikaans vliegveld. Op Schiphol gaat het er iets moderner aan toe: de verkeerstoren kan via camerabeelden de status van de gate checken. Toch moet dat met data een stuk efficiënter kunnen, vonden Peironcely en zijn team.

'Maar wij zijn geen eigenaar van de data', vertelde Peironcely. 'Die is in handen van KLM, het vrachtbedrijf of andere bedrijven in het ecosysteem.' Het team besloot daarop om de videofeed van de camera's te gebruiken om de bedrijvigheid in kaart te brengen. Het idee was om de objecten in de video als indicatoren te gebruiken. De aanwezigheid van een cateringwagen of brandstoftruck toont welke activiteit wordt uitgevoerd. Maar zo makkelijk ging dat niet.

Googles kattenplaatjes en de werkstudent

Ook de camera's zijn niet van Schiphol, die zijn van de beveiliging. Het duurde vier maanden voor het team de vergunningen had om de videofeed te gebruiken. Vervolgens moest elk object op elk frame van de video geïdentificeerd worden. Peironcely: 'De voorbeelden van beeldherkenning van Google en hun kattenplaatjes zien er erg indrukwekkend uit. Maar als je een beeldherkenningssysteem op ruwe video loslaat, dan werkt het voor geen meter. We hebben een student ingehuurd die twee weken lang handmatig alle objecten heeft gelabeld.'

De plaatjes met handmatige labels werden vervolgens ingezet om een neuraal netwerk te trainen. Van het voorbeeldmateriaal leerde de computer patronen herkennen. Op basis van die patronen deed de computer voorspellingen over nieuwe plaatjes. 'We controleerden handmatig de kwaliteit van de voorspellingen,' zei Peironcely. 'Verkeerde voorspellingen werden terug in het model ingevoerd om het neurale netwerk opnieuw te trainen.'

De samenwerking tussen mens en machine leidde uiteindelijk tot een weergave van alle objecten op een tijdlijn. 'Je krijgt een gigantische grafiek waarop je kan zien wanneer elke activiteit begint en eindigt', zei Peironcely. 'Als je heel veel van deze data hebt, kan je zien welke activiteiten efficiënter kunnen.'

Om KI in de praktijk toe te passen is data nodig. Heel veel bruikbare, correcte data. Het relaas van Peironcely laat zien dat het verkrijgen van bestaande data moeilijk kan zijn omdat die van andere partijen is. Ook het creëren van data is een omslachtig proces. En dat is niet de enige moeilijkheid waar het datateam van Schiphol mee kampt. Ook nadat een

toepassing is ontwikkeld, zijn er hindernissen. Peironcely: 'Voor de implementatie van een applicatie moeten meerdere partijen bereid zijn mee te doen: vliegmaatschappijen, bagage-afhandelaars, beveiliging. Voor de meeste van hen heeft veiligheid de hoogste prioriteit. Ze zijn daarom terughoudend om procedures die goed werken te veranderen. Daarom moeten we veel met hen praten, zaken goed uitleggen en geduld hebben.'

Bejaarde computers in de medische wereld

Uit een paneldiscussie over de toepassingen van KI in de gezondheidszorg komt eenzelfde beeld naar voren. Theoretisch kan KI tot enorme verbeteringen leiden, maar de praktijk is weerbarstig. Milan Petković, hoofd Data Science bij Philips Healthcare, zou graag zien dat de aandacht in de gezondheidszorg verschuift van genezen naar voorkomen. KI zou daarbij kunnen helpen, bijvoorbeeld door lichaamsfuncties als hartslag en bloeddruk te monitoren met draagbare sensoren. Of door grote, langdurige onderzoeken te doen naar het effect van eetgewoonten op de gezondheid. 'Maar daarvoor is heel veel data nodig en dat is een uitdaging', zei Petković.

Het delen van data in de medische sector is lastig. Het gaat om gevoelige, persoonlijke data die niet zomaar gedeeld mag worden. Daarnaast zijn de databronnen gefragmenteerd. Petković: 'Het landschap is gedecentraliseerd. Er zijn veel verschillende partijen waardoor de data in silo's belandt.' Als die partijen al overeenkomen hun data te delen dan zijn er technische obstakels. 'Interoperabiliteit is een uitdaging', zei Petković. 'In de gezondheidszorg zijn veel ouderwetse ICT-systemen. Die kunnen niet zo maar op elkaar worden aangesloten. Ook voldoen die oude systemen vaak niet aan de hedendaagse privacy- en veiligheidsnormen waardoor ze niet kunnen worden opengesteld voor derden.'

De inzichten van Petković maken — net als het praktijkvoobeeld van Peironcely — duidelijk dat het toepassen van KI op bestaande systemen en processen veel voeten in de aarde heeft. ◄

180462-01

Weblinks

[1] www.ai-expo.net/europe/collateral/



Momentopname van het omdraaiproces op vliegveld Gardermoen, Oslo. Foto: Trond Kvitvik (Creative Commons BY 3.0-licentie).



welkom in je WEB STORE

TIP VAN DE REDACTIE



De YDLidar X4 is voornamelijk bedoeld voor toepassingen bij robots en maakt gebruik van een pulserende laser om de afstand tot objecten te meten. De laser reflecteert op het oppervlak van het object en door de tijd te meten die het licht erover doet om op de sensor te vallen wordt de afstand tot het object bepaald. De YDLidar X4 maakt gebruik van een roterende kop waarin zowel de laser als de ontvanger zijn gemonteerd.



Met deze roterende kop kan het systeem 360º om zich heen 'kijken'. Wel alleen maar in één vlak, dat spreekt voor zich. Met de YDLidarX4 krijg je voor een zeer redelijk bedrag een prima bruikbare Lidar waarmee je je robot ogen in alle richtingen geeft en die ook nog eens met een flinke berg software komt.

Thijs Beckers (Redactie Elektor)

www.elektor.nl/ydlidar-x4

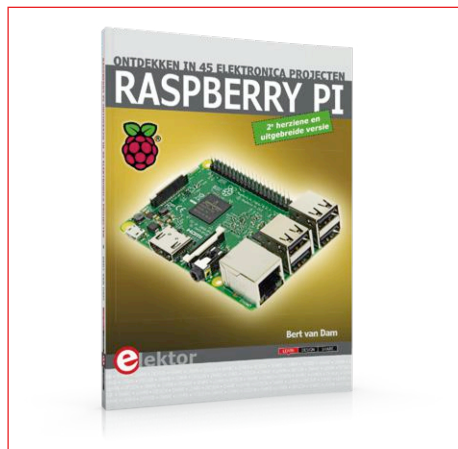
Elektor Bestsellers

1. RasPad RPi-tablet
   www.elektor.nl/raspad



2. ESP32 programmeren voor beginners
   www.elektor.nl/esp32-programmeren
3. Raspberry Pi 3 (model B+)
   www.elektor.nl/rpi3b-plus
4. Nixie Bargraph Thermometer
   www.elektor.nl/nixie-thermometer
5. Raspberry Pi 3 Basic to Advanced Projects
   www.elektor.nl/rpi3-projecten
6. Walabot Creator Pack
   www.elektor.nl/walabot
7. pi-top 2 RPi-Laptop-Kit
   www.elektor.nl/pi-top-v2

Raspberry Pi Ontdekken



In dit boek staan de sterke kanten van de Raspberry Pi centraal: de combinatie van programmeren en elektronica. Na een korte introductie worden maar liefst 45 spannende en interessante projecten in detail besproken en uitgelegd. Van een wisselknipperlicht tot lichtmeter en temperatuurregeling en van een motorsnelheidsregeling tot een webserver met CGI.

 Ledenprijs: € 35,96

www.elektor.nl/rpi-ontdekken

Programming with STM32 Nucleo Boards

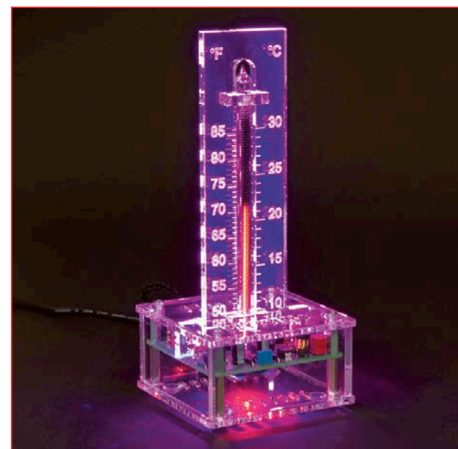


Dit (Engelstalige) boek bevat projecten met de STM32 Nucleo ontwikkelboards van STMicroelectronics. De projecten variëren van eenvoudige knipperende LED's tot complexere meer projecten. De meeste zijn gebaseerd op het Nucleo-L476RG board. Dit board (t.w.v. € 15) krijgt u tijdelijk gratis bij aankoop van het boek!

 Ledenprijs: € 31,46

www.elektor.nl/stm32-boek

Nixie Bargraph Thermometer



Deze Nixie Bargraph Thermometer is een thermometer met een traditionele schaal in graden C/F maar met een "warme" weergave. Een perfecte match tussen klassiek en modern en bovendien een juweeltje voor in de woonkamer. De Nixie Bargraph Thermometer maakt gebruik van de Russische IN-9, en is weer eens iets anders dan een traditioneel klokje.

 Ledenprijs: € 62,96

www.elektor.nl/nixie-thermometer

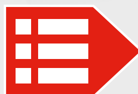




Android App Development

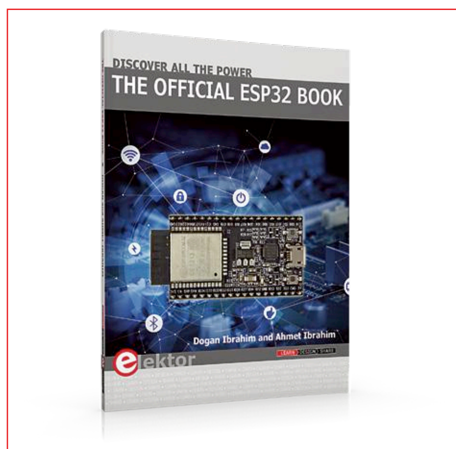
With Basic for Android – B4A

Apps ontwikkelen om elektronica met je smartphone aan te sturen wordt steeds populairder. In dit Engelstalige boek presenteert Dogan Ibrahim de programmeertaal Basic For Android (B4A) en laat hij zien hoe je deze in eenvoudige elektronicaprojecten kunt toepassen. Het boek beschrijft hoe een Android mobiel apparaat kan communiceren met populaire hardware platforms (waaronder Raspberry Pi, Arduino, en ESP32) via een Wi-Fi verbinding of door gebruik te maken van SMS tekstberichten. Het boek bevat veel voorbeelden en projecten die naar eigen behoefte aangepast kunnen worden.



LEDENPRIJS: € 24,95
www.elektor.nl/android-app-development

The Official ESP32 Book

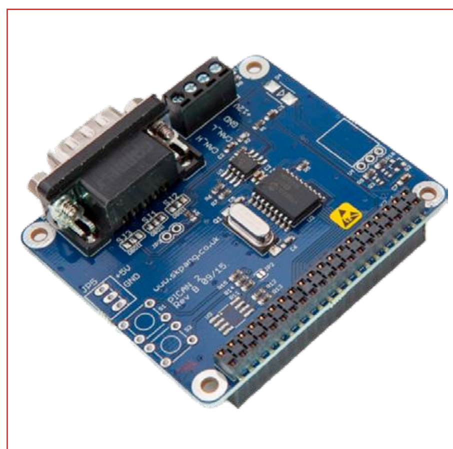


Dit (Engelstalige) boek is een introductie van de ESP32 microcontroller van Espressif Systems en beschrijft de belangrijkste hardware- en software functies van deze chip. De nieuwe ESP32 heeft dezelfde functionaliteiten als de populaire ESP8266 microcontroller, maar is uitgebreid met onder andere bluetooth, hogere CPU-snelheid, sensoren, busverbindingen en beveiligingsfuncties. In het boek leer je hoe je ESP32 hardware en software gebruikt in praktische projecten.

 Ledenprijs: € 31,46

www.elektor.nl/esp32-book

PiCan 2 – CAN-bus-board voor Raspberry Pi



Een PiCAN 2 HAT geeft uw Raspberry Pi volledige CAN-Bus-functionaliteit: CAN v2.0B bij 1 Mb/s en snelle (10 MHz) SPI -interface. Hij werkt met een Microchip MCP2515 CAN-controller en een MCP2551 CAN-transceiver. De verbinding wordt gemaakt via een 9-polige D-connector of via een 3-polige schroefconnector. De SocketCAN-driver is gemakkelijk te installeren en programmeren kan in Python of in C.

 Ledenprijs: € 35,96

www.elektor.nl/pi-can2

ESP32 programmeren voor beginners



In dit boek bekijken we de ESP32 en zijn voorganger, de ESP8266. Deze kleine, maar zeer krachtige Chinese chips bestaan uit een processor, geheugen, input- en outputaansluitingen en hebben draadloze netwerktoegang via WiFi. Met dit boek tonen we hoe de programmeeromgeving Arduino werkt en hoe die gebruikt kan worden met ESP-hardware. Aan de hand van voorbeelden en projecten verduidelijken we hoe de ESP-hardware kan worden toegepast.

 Ledenprijs: € 20,25

www.elektor.nl/esp32-programmeren

Hexadoku puzzelen voor elektronici

PC, oscilloscoop en soldeerbout kunnen weer even op adem komen terwijl u uw hersenen pijnigt met onze Hexadoku. De instructies voor deze puzzel zijn heel eenvoudig. De Hexadoku werkt met de hexadecimale getallen 0 t/m F, helemaal in de stijl van elektronici en programmeurs.

Vul het diagram van 16 x 16 hokjes zodanig in dat **alle** hexadecimale cijfers van 0 t/m F (dus 0...9 en A...F) precies éénmaal voorkomen in elke rij, in elke kolom en in elk vak van 4 x 4 hokjes (gemarkeerd door de dikkere zwarte lijnen).
Een aantal cijfers is al aangegeven en deze bepalen de uitgangssituatie voor de puzzel.
Onder de inzenders met de goede oplossing verloten we vijf waardebonnen. Om mee te dingen naar een van deze prijzen dient u **de cijfers in de grijze hokjes** naar ons op te sturen.



Doe mee en win! Onder de internationale inzenders met het juiste antwoord verloten we **vijf Elektor-waardebonnen**, elk ter waarde van **50 Euro**.
Het is dus zeker de moeite waard om mee te doen!

Stuur uw antwoord (de getallen in de grijze hokjes) **vóór 29 november 2018** naar: **hexadoku@elektor.nl**

De prijswinnaars

De juiste oplossing van de Hexadoku uit het september/oktober-nummer is: **D83E1**.
De Elektor-waardebonnen van 50 Euro zijn gewonnen door Manfred Häfner (Duitsland), Olli Hakala (Finland), Axel Stefanini (Frankrijk), Brian Wood (Groot-Brittannië) en Alexandr Papazyan (Rusland).
Allemaal van harte gefeliciteerd!

		3		F	9					6	A	5		C	
F								7		8		E	6	9	4
		5	8	1			B		9					7	A
	B	C	D				7			F					3
							C	5	1		4		7		
		9				0	E	3		D	2			1	F
D		0	5	3	7		2		6				4		
	1		4		8					0				6	
	4				D					1		6		2	
		D				8		F		C	7	0	A		E
8	0			7	E		4	9	B				C		
		A		0		2	1	D							
5					1			6				2	B	3	
2	A					3		0			8	7	E		
3	8	4	0		A		6								9
	D		E	2	4					9	F		1		

5	F	1	C	A	0	6	D	4	7	8	3	9	B	2	E
9	D	8	3	E	1	F	7	5	A	B	2	6	C	0	4
E	A	0	7	2	8	4	B	D	6	9	C	3	F	1	5
2	4	B	6	C	3	5	9	E	F	0	1	7	8	D	A
F	B	2	5	0	D	7	C	6	E	3	8	A	1	4	9
3	0	6	D	4	F	2	A	9	1	C	7	E	5	8	B
4	7	9	8	3	E	1	5	A	B	2	0	F	6	C	D
A	C	E	1	6	9	B	8	F	4	D	5	0	2	3	7
C	E	4	A	1	2	0	6	3	D	5	B	8	7	9	F
6	3	7	9	F	5	8	E	C	0	A	4	2	D	B	1
8	5	D	B	9	7	A	4	1	2	F	6	C	0	E	3
0	1	F	2	B	C	D	3	7	8	E	9	5	4	A	6
B	6	C	4	7	A	E	2	0	3	1	F	D	9	5	8
1	8	5	E	D	6	9	0	B	C	7	A	4	3	F	2
D	9	3	F	8	4	C	1	2	5	6	E	B	A	7	0
7	2	A	0	5	B	3	F	8	9	4	D	1	E	6	C